

Chapter 8

Implementation Support

Elements of windowing systems

◦ Windowing System:

- windowing system is a system for sharing a computer's graphical display resources/ GUI among multiple applications at the same time.

0 نظام النوافذ:

0 نظام النوافذ هو نظام لمشاركة موارد العرض الرسومية / واجهة المستخدم الرسومية للكمبيوتر بين تطبيقات متعددة في نفس الوقت.

◦ Roles of a windowing system:

- Device/Hardware Independent
- Management of independent but simultaneously active applications

0 أدوار نظام النوافذ:

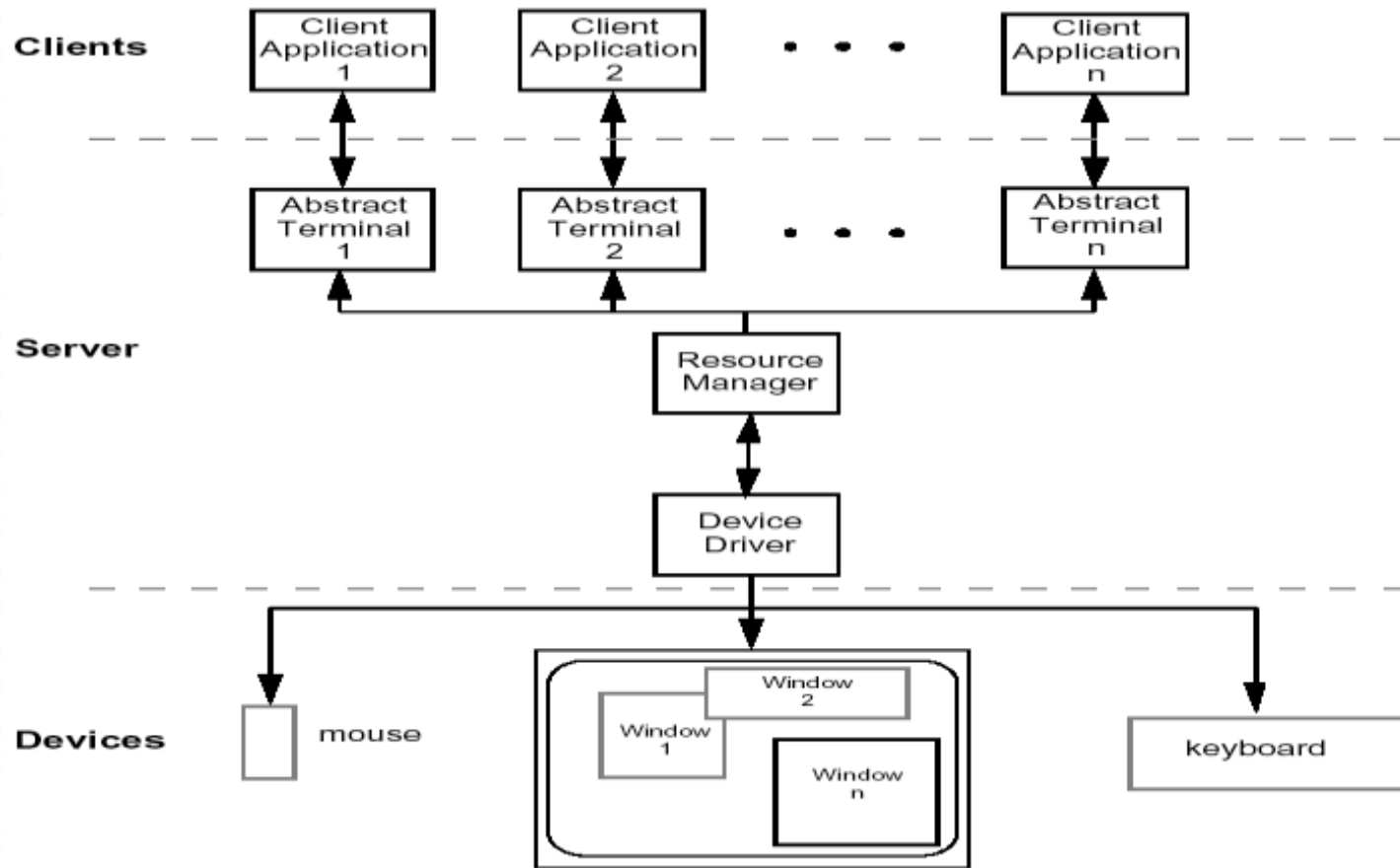
0 جهاز / أجهزة مستقلة

0 إدارة تطبيقات مستقلة ولكن في نفس الوقت نشط

Elements of windowing systems

- 0 من الضروري أن تكون قادرًا على برمجة تطبيق يعمل على نطاق واسع من الأجهزة.
- 0 It is imperative to be able to program an application that will run on a wide range of devices. 0 يحتاج المبرمج إلى توجيه الأوامر إلى محطة مجردة - يفهم لغة أكثر عمومية ويمكن ترجمتها إلى لغة العديد من الأجهزة المحددة الأخرى.
- 0 Programmer needs to direct commands to an abstract terminal - understands a more generic language and can be translated to the language of many other specific devices. 0 محطة الملخص: 0 قابلية تطبيق البرنامج ممكن ، 0 يحتاج برنامج واحد فقط - برنامج تشغيل الجهاز - إلى كتابته لجهاز معين ، 0 Abstract terminal: ومن ثم يمكن لأي برنامج تطبيق الوصول إليه.
- 0 Portability of application program possible,
- 0 Only one program – device driver – needs to be written for a particular hardware device and then any application program can access it.
- 0 هل لديك لغة عامة ثابتة لمحطة مجردة والتي تسمى نموذج التصوير.
- 0 Have fixed generic language for the abstract terminal which is called its imaging model.

Windowing System Architecture



The client-server architecture

PROGRAMMING THE APPLICATION

0 التطبيقات التفاعلية عادة ما تكون مدفوعة بالمستخدم بمعنى أن الإجراء الذي يتخذه التطبيق يتم تحديده من خلال المدخلات الواردة من المستخدم.

- Interactive applications are generally user driven in the sense that the action the application takes is determined by the input received from the user.

0 يمكن استخدام نموذجين للبرمجة لتنظيم تدفق التحكم داخل التطبيق.

- Two programming paradigms can be used to organize the flow of control within the application.

- The windowing system does not necessarily determine which of these two paradigms is to be followed.

- Read-evaluation loop

- Notification based

0 لا يحدد نظام النوافذ بالضرورة أي من هذين النموذجين يجب اتباعه.

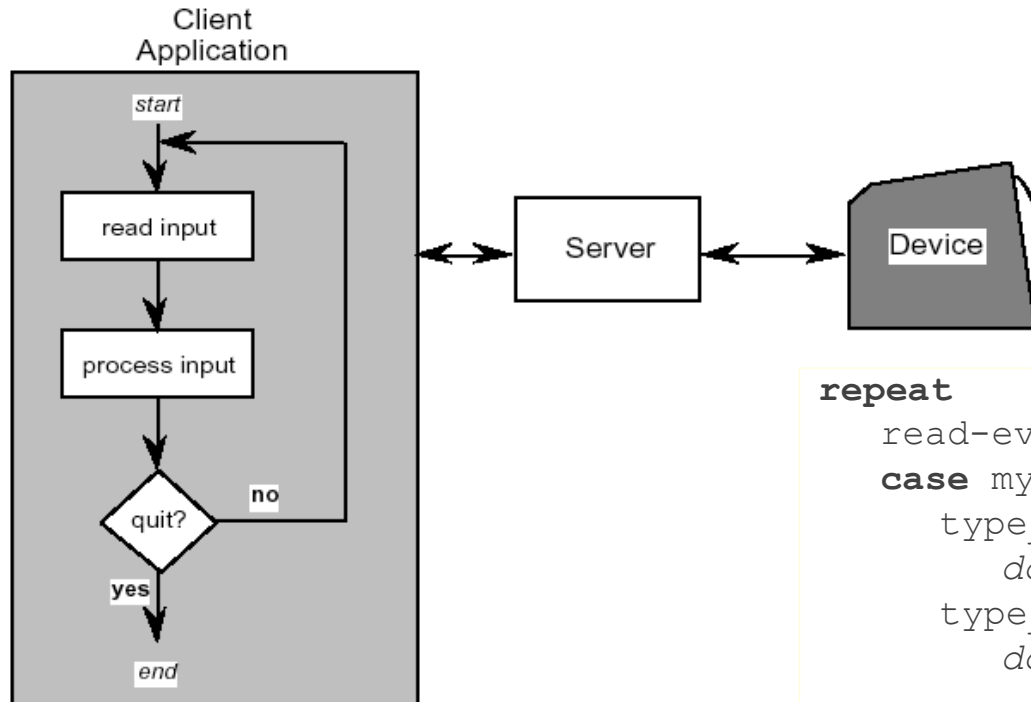
0 حلقة للقراءة

0 يستند الإخطار

PROGRAMMING THE APPLICATION

read-evaluation loop

برمجة التطبيق
حلقة تقييم للقراءة



```
repeat
  read-event (myevent)
  case myevent.type
    type_1:
      do type_1 processing
    type_2:
      do type_2 processing
    ...
    type_n:
      do type_n processing
  end case
end repeat
```

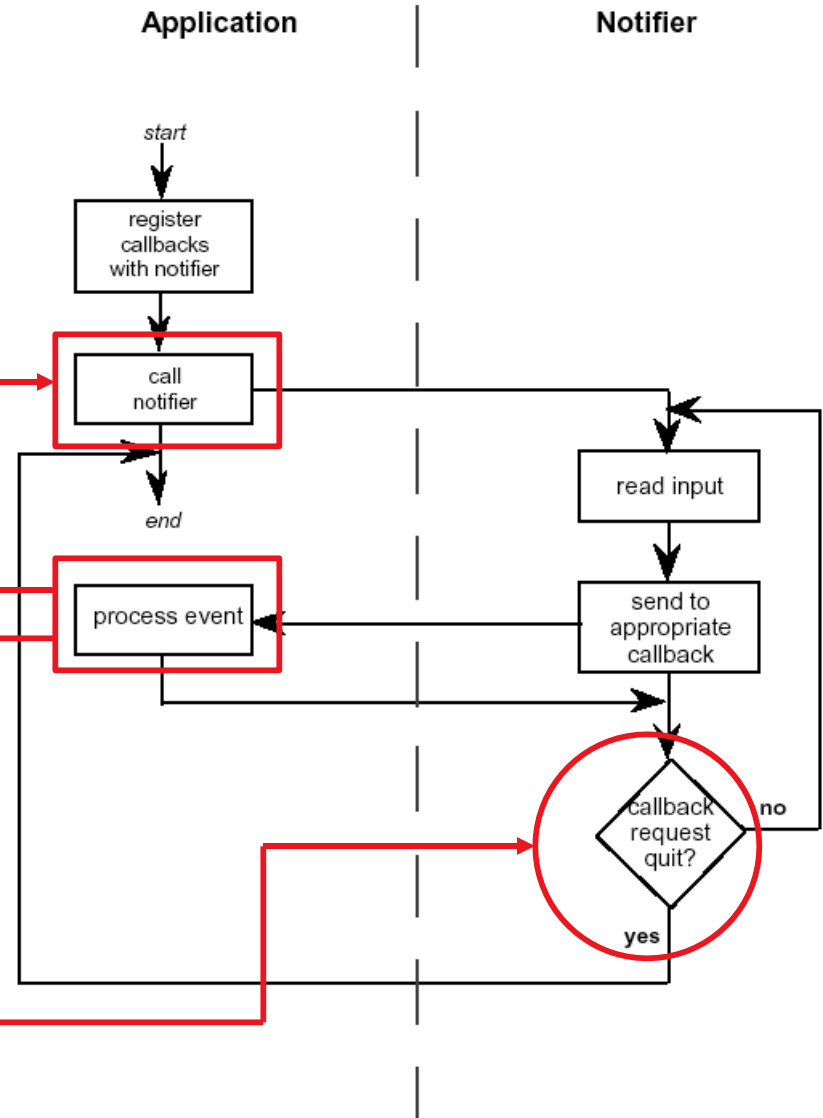
PROGRAMMING THE APPLICATION

Notification-based

```
void main(String[] args) {  
    Menu menu = new Menu();  
    menu.setOption("Save");  
    menu.setOption("Quit");  
    menu.setAction("Save",mySave)  
    menu.setAction("Quit",myQuit)  
    ...  
}
```

```
int mySave(Event e) {  
    // save the current file  
}
```

```
int myQuit(Event e) {  
    // close down  
}
```



0 توفر مجموعة الأدوات للمبرمج مجموعة من كائنات التفاعل الجاهزة - والتي تسمى أساليب التفاعل والأدوات
0 هذه الكائنات لها سلوك محدد مسبقًا كما هو موضح في الزر:

Toolkits

- 0 A toolkit provides the programmer with a set of ready-made interaction objects – alternatively called interaction techniques, gadgets
- 0 These objects have a predefined behavior as that described for the button:

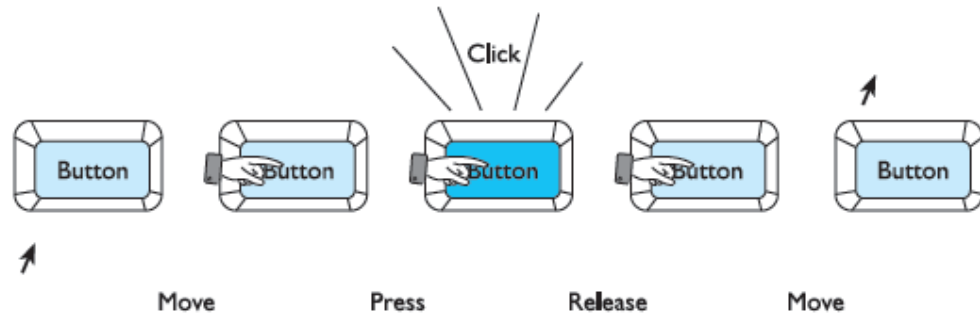


Figure 8.8 Example of behavior of a button interaction object

- 0 The Java toolkit for developing windowed applications is called the Abstract Windowing Toolkit, AWT

0 يُطلق على مجموعة أدوات Java لتطوير التطبيقات ذات الإطارين اسم مجموعة أدوات تجريد الإطارات المجردة ،
AWT

Toolkits (ctd)

- Toolkits provide Consistency and Generalizability for an interactive system.
- One of the advantages of programming with toolkits is that they can enforce consistency in both input form and output form by providing similar behavior to a collection of widgets
- This consistency of behavior for interaction objects is referred to as the look and feel of the toolkit

توفر الأدوات مجموعة الاتساق والتعميم لنظام تفاعلي.

0 واحدة من مزايا البرمجة مع مجموعة الأدوات هي أنها تستطيع فرض الاتساق في كل من نموذج الإدخال والإخراج من خلال توفير سلوك مماثل لمجموعة من الأدوات
يشار إلى هذا الاتساق في سلوك كائنات التفاعل على أنه الشكل والمظهر في مجموعة الأدوات

Toolkits

◦ To provide flexibility, the interaction objects can be modified

◦ These objects are modified by -----

◦ *Instantiation?*

◦ *Inheritance*

◦ *multiple inheritance?*

◦ *instance attributes?*

0 لتوفير المرونة ، يمكن تعديل كائنات التفاعل
0 يتم تعديل هذه الكائنات بواسطة -----

0 التهيئة؟

0 الوراثة

0 ميراث متعدد؟

0 سمات سبيل المثال؟

User Interface Management Systems (UIMS)

0 مجموعة تقنيات البرمجة والتصميم التي توفر المزيد من دعم التطوير لتصميم النظام التفاعلي خارج مجموعة الأدوات.

- 0 The set of programming and design techniques which provide more development support for interactive system design beyond the toolkits.

0 من أمثلة UIMS هي Serpent & Picasso

- 0 Examples of UIMS are Serpent & Picasso

0 يجب أن يدعم UIMS:

- 0 The UIMS should support:

0 معمارية مفاهيمية:

- 0 Conceptual Architecture:
 - 0 من أجل بنية نظام تفاعلي يركز على الفصل بين دلالات التطبيقات / المنطق والعرض ؛

- 0 for the structure of an interactive system which concentrates on a separation between application semantics/logics and presentation;

- 0 Techniques:

- 0 for implementing a separated application and presentation and preserving the intended connection between them;

0 تقنيات:

- 0 Support techniques: 0 لتنفيذ تطبيق منفصل وعرضه والحفاظ على الصلة المقصودة بينهما ؛

- 0 For managing, implementing and evaluating a run-time interactive environment

0 تقنيات الدعم:

0 لإدارة وتنفيذ وتقييم بيئة تفاعلية وقت التشغيل

UIMS: Conceptual Architecture

الفصل بين دلالات التطبيق / المنطق والعرض التقديمي:

◦ Separation between application semantics/logic and presentation improving:

0 النقل

◦ **Portability**

0 يعمل على أنظمة مختلفة ويوفر واجهة مستقلة للجهاز

◦ runs on different systems and provides device independent interface

0 إعادة القوام

◦ **Reusability**

0 إعادة استخدام المكونات تقلل من تكاليف التطوير

◦ Reusability of components reduces development costs

◦ **Multiple interfaces**

◦ Supports development of multiple interface to access same functionality

◦ **Customizability**

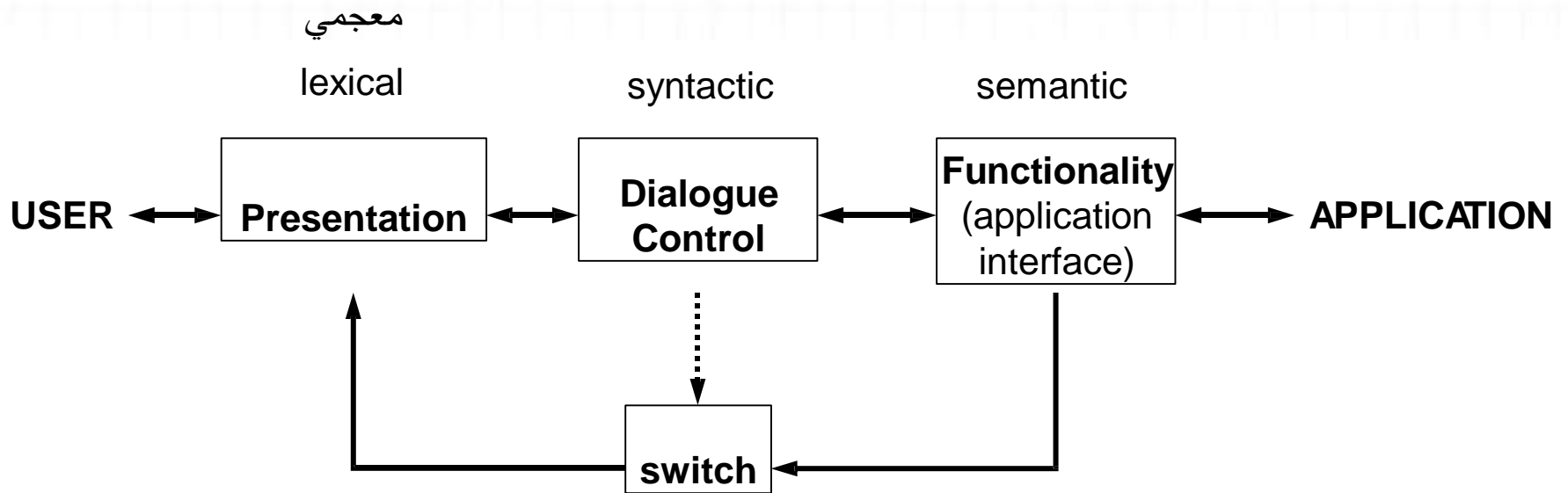
◦ by designer and user without altering core application واجهات متعددة

0 يدعم تطوير واجهة متعددة للوصول إلى الوظائف

0 التخصيص

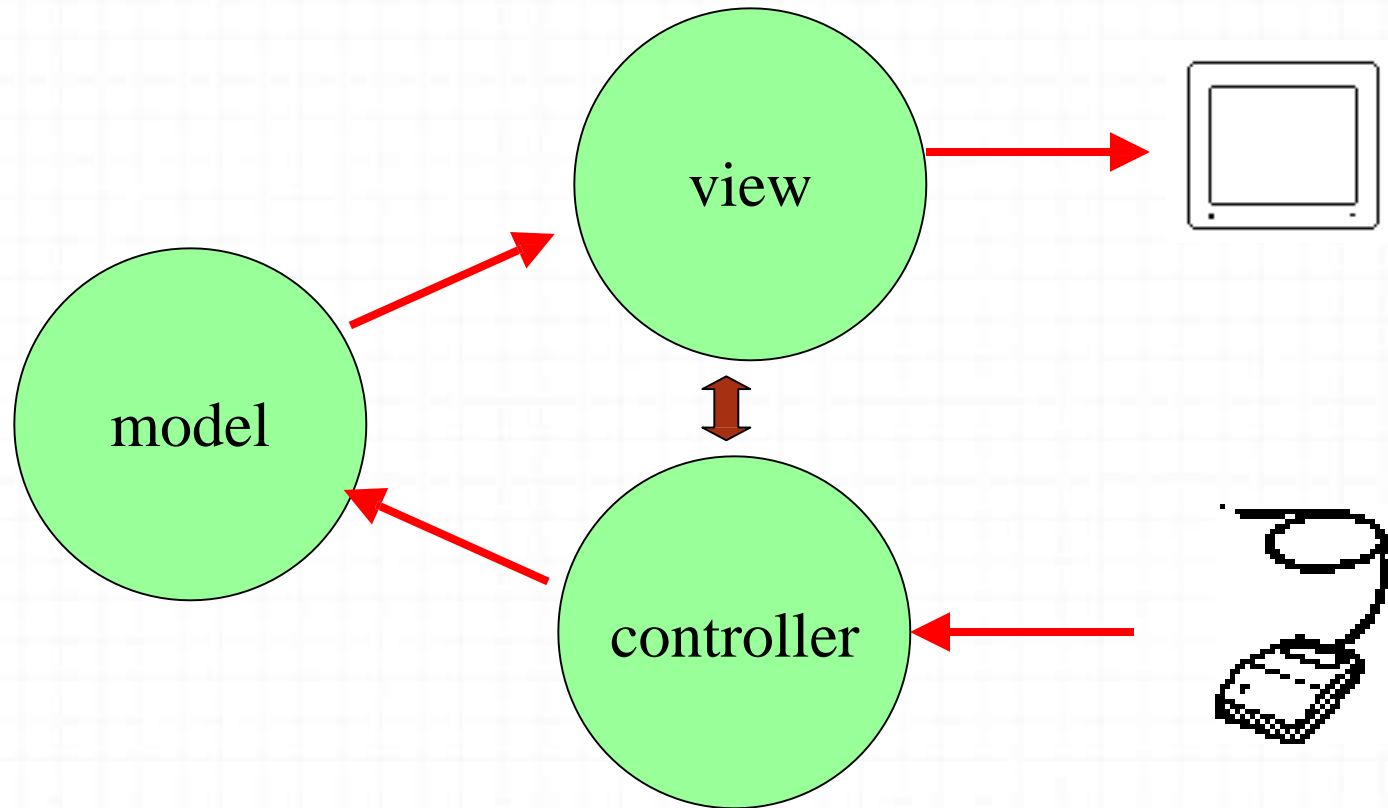
0 من المصمم والمستخدم دون تغيير التطبيق الأساسي

UIMS: Conceptual Architecture



Seeheim Model

UIMS: Conceptual Architecture



MVC: model - view - controller

UIMS: Conceptual Architecture

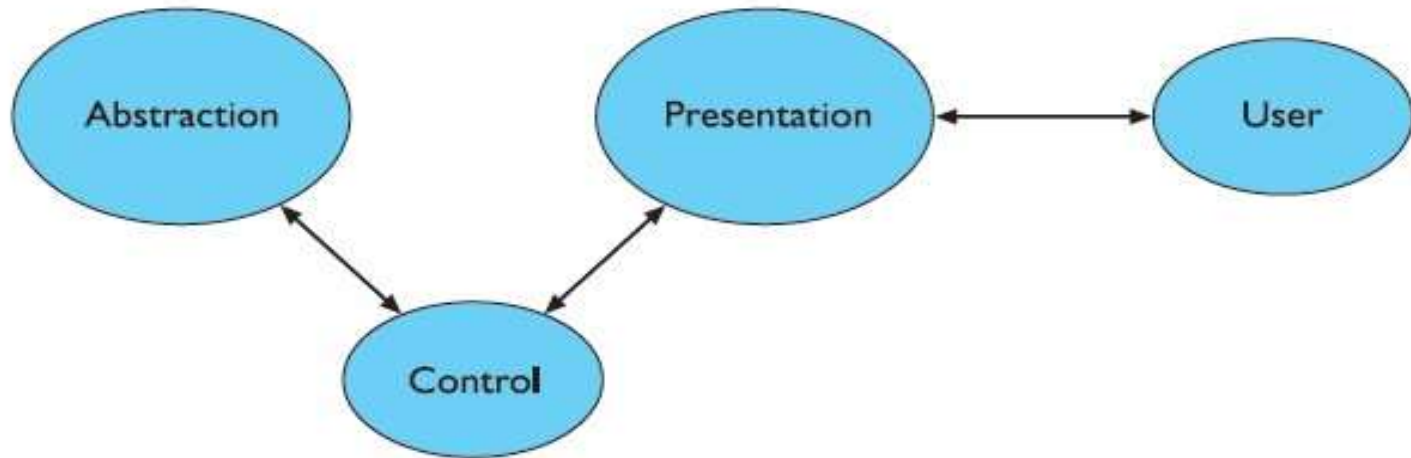


Figure 8.12 The presentation–abstraction–control model of Coutaz

Multi-Agent Architecture

PAC

presentation - abstraction - control

العمارة متعددة الوكيل

PAC

عرض - التجريد - السيطرة

MVC & PAC Differences/Issues

المجموعات المدخلات والمخرجات معا.

ثانياً ، يوفر PAC عنصراً صريحاً يتمثل واجبه في رؤية التجريد والعرض التقديمي متناسقين مع بعضهما البعض.

PAC

MVC

- Groups input and output together.
- Secondly, PAC provides an explicit component whose duty it is to see that abstraction and presentation are kept consistent with each other.
- Not linked to any programming environment, though it is certainly helpful to an object-oriented approach.

It is probably because of this last difference that PAC could so easily isolate the control component; PAC is more of a conceptual architecture.

غير مرتبط بأي بيئة برمجة ، على الرغم من أنه بالتأكيد مفيد لنهج موجه إلى الهدف.

ربما يرجع هذا الاختلاف الأخير إلى قدرة PAC على عزل عنصر التحكم بسهولة ؛ PAC هو أكثر من بنية المفاهيم.

- whereas MVC separates them
- MVC does not assign this important task to any one component, leaving it to the programmer/designer to determine where that chore resides
- Cannot so easily isolate the control component

MVC

في حين أن MVC تفصلها

لا تقوم شركة MVC بتخصيص هذه المهمة المهمة لأي مكون واحد ، تاركة الأمر للمبرمج / المصمم لتحديد مكان وجود هذا العمل الرتيب
لا يمكن عزل مكون التحكم بسهولة

اعتبارات التنفيذ

Implementation Considerations

Dialog Modeling Techniques in UIMS:

- Menu Networks:

- Grammar notations

- State transition diagrams

- Event Languages

- Declarative languages

- Constraints

- Graphical specification

0 تقنيات نمذجة الحوار في UIMS:

0 شبكات القائمة:

0 تدوين القواعد

0 المخططات الانتقالية للدولة

0 لغات الحدث

0 اللغات التعريفي

0 القيود

0 المواصفات الرسومية