

人机交互技术

施展

华中科技大学 武汉光电国家实验室

October 22, 2012



内容提要

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 1 第六讲
 - 人机交互界面表示模型
 - 界面描述语言
 - 窗口系统
 - 用户界面管理系统

- 2 小结

第六讲 人机交互界面表示模型与实现

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 目的: 在界面设计的早期阶段, 研究建立一种用户界面表示模型
 - 利用形式化的设计语言来分析和表达用户任务以及用户和系统之间的交互情况;
 - 使界面表示模型能方便地映射到实际的设计实现。

人机交互界面表示模型

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

本节介绍人机交互界面主要表示模型及其转换

■ 行为模型：

该模型主要从用户和任务的角度考虑如何来描述人机交互界面。

■ 结构模型：

该模型主要从系统的角度来表示人机交互界面。

■ 模型转换：

主要介绍行为模型到结构模型的转换。

■ 表现模型：

主要介绍人机界面表现的具体描述方法。

人机交互界面表示模型 行为模型

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- GOMS: Goal, Operator, Method, Selection
- LOTOS: Language Of Temporal Ordering Specification
- UAN: User Action Notation

人机交互界面表示模型 GOMS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 1983年由Card, Morgan和Newell 提出的。
- 通过目标 (Goal)、操作 (Operator)、方法 (Method) 以及选择规则 (Selection) 四个元素来描述用户的行为。
- GOMS是在交互系统中用来分析、建立用户行为的模型，采用“分而治之”的思想，将一个任务进行多层次的细化。

人机交互界面表示模型 GOMS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 目标 Goals

- 目标就是用户执行任务最终想要得到的结果,它可以在不同的层次中进行定义。
- 例: “编辑一篇文章” - “编辑文章” (高层); “删除字符” (低层)

■ 操作 Operators

- 操作是任务分析到最低层时的行为, 是用户为了完成任务所必须执行的基本动作。
- 操作不能被分解, 在GOMS模型中是原子动作。

人机交互界面表示模型 GOMS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 目标 Goals

- 目标就是用户执行任务最终想要得到的结果,它可以在不同的层次中进行定义。
- 例: “编辑一篇文章” - “编辑文章” (高层); “删除字符” (低层)

■ 操作 Operators

- 操作是任务分析到最低层时的行为, 是用户为了完成任务所必须执行的基本动作。
- 操作不能被分解, 在GOMS模型中是原子动作。

人机交互界面表示模型 GOMS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 方法 Methods

- 方法是描述如何完成目标的过程。
- 一个方法本质上来说是内部的算法，用来确定子目标序列及完成目标所需要的操作。

■ 选择 Selection

- 选择是用户要遵守的判定规则，以确定在特定环境下所使用的方法。
- 当有多个方法可供选择时，GOMS中并不认为这是一个随机的选择，而是尽量来预测会使用哪个方法,这需要根据特定用户、系统的状态、目标的细节来预测要选择哪种方法。

人机交互界面表示模型 GOMS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 例：窗口最小化

GOAL: ICONISE-WINDOW

```
. [select GOAL: USE-CLOSE-METHOD
    . MOVE-MOUSE-TO-WINDOW-HEADER
    . POP-UP-MENU
    . CLICK-OVER-CLOSE-OPTION
GOAL: USE-L7-METHOD
    . PRESS-L7-KEY]
```

For a particular user:

Rule 1: Select USE-CLOSE-METHOD unless another rule applies

Rule 2: If the application is GAME, select L7-METHOD

人机交互界面表示模型 GOMS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 例：窗口最小化

Six execution phase operators

Physical motor

K - key stroking

P - pointing

H - homing

B - button pressing

Mental

M - mental preparation

System

R - response

Times are empirically determined ($T=Task$).

$$T_{\text{execute}} = T_K + T_P + T_H + T_B + T_M + T_R$$

人机交互界面表示模型 GOMS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 例：窗口最小化

assume hand starts on mouse

USE-L7-METHOD		USE-CLOSE-METHOD	
H[to keyboard]	0.40	P[to menu]	1.1
M	1.35	B[LEFT down]	0.1
K[L7 key]	0.28	M	1.35
		P[to option]	1.1
		B[LEFT up]	0.1
Total	2.03 secs	Total	3.75 secs

人机交互界面表示模型 GOMS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 例：删除文件（命令行）

- **Goal:** Delete a File
- **Method** for accomplishing goal of deleting a file
 - retrieve from Long term memory that command verb is "del"
 - think of directory name & file name and make it the first listed parameter
 - accomplish goal of entering & executing command
 - return with goal accomplished

人机交互界面表示模型 GOMS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 例：删除文件（图形界面）

- **Goal:** Delete a File
- **Method** for accomplishing goal of deleting a file
 - find file icon
 - accomplish goal of dragging file to trash
 - Return with goal accomplished

人机交互界面表示模型 GOMS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 是一种人机交互界面表示的理论模型，被称为最成熟的工程典范 该模型在计算机系统的评估方面也有广泛的应用。
- 没有清楚的描述错误处理的过程， 假设用户完全按一种正确的方式进行人机交互， 因此只针对那些不犯任何错误的专家用户。
- GOMS对于任务之间的关系描述过于简单， 只有顺序和选择， 事实上任务之间的关系还有很多种。
- 选择关系通过非形式化的附加规则描述， 实现起来比较困难。
- 把所有的任务都看作是面向操作目标的， 而忽略了一些任务所要解决的问题本质以及用户间的个体差异， 它的建立不是基于现有的认知心理学， 无法代表真正的认知过程。

人机交互界面表示模型 GOMS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 是一种人机交互界面表示的理论模型，被称为最成熟的工程典范 该模型在计算机系统的评估方面也有广泛的应用。
- 没有清楚的描述错误处理的过程， 假设用户完全按一种正确的方式进行人机交互，因此只针对那些不犯任何错误的专家用户。
- GOMS对于任务之间的关系描述过于简单， 只有顺序和选择，事实上任务之间的关系还有很多种。
- 选择关系通过非形式化的附加规则描述，实现起来比较困难。
- 把所有的任务都看作是面向操作目标的，而忽略了一些任务所要解决的问题本质以及用户间的个体差异， 它的建立不是基于现有的认知心理学，无法代表真正的认知过程。

人机交互界面表示模型 GOMS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 是一种人机交互界面表示的理论模型，被称为最成熟的工程典范 该模型在计算机系统的评价方面也有广泛的应用。
- 没有清楚的描述错误处理的过程， 假设用户完全按一种正确的方式进行人机交互， 因此只针对那些不犯任何错误的专家用户。
- GOMS对于任务之间的关系描述过于简单， 只有顺序和选择，事实上任务之间的关系还有很多种。
- 选择关系通过非形式化的附加规则描述，实现起来比较困难。
- 把所有的任务都看作是面向操作目标的，而忽略了一些任务所要解决的问题本质以及用户间的个体差异， 它的建立不是基于现有的认知心理学，无法代表真正的认知过程。

人机交互界面表示模型 GOMS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 是一种人机交互界面表示的理论模型，被称为最成熟的工程典范 该模型在计算机系统的评估方面也有广泛的应用。
- 没有清楚的描述错误处理的过程， 假设用户完全按一种正确的方式进行人机交互， 因此只针对那些不犯任何错误的专家用户。
- GOMS对于任务之间的关系描述过于简单， 只有顺序和选择，事实上任务之间的关系还有很多种。
- 选择关系通过非形式化的附加规则描述，实现起来比较困难。
- 把所有的任务都看作是面向操作目标的，而忽略了一些任务所要解决的问题本质以及用户间的个体差异， 它的建立不是基于现有的认知心理学，无法代表真正的认知过程。

人机交互界面表示模型 GOMS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 是一种人机交互界面表示的理论模型，被称为最成熟的工程典范 该模型在计算机系统的评估方面也有广泛的应用。
- 没有清楚的描述错误处理的过程， 假设用户完全按一种正确的方式进行人机交互， 因此只针对那些不犯任何错误的专家用户。
- GOMS对于任务之间的关系描述过于简单， 只有顺序和选择，事实上任务之间的关系还有很多种。
- 选择关系通过非形式化的附加规则描述，实现起来比较困难。
- 把所有的任务都看作是面向操作目标的，而忽略了一些任务所要解决的问题本质以及用户间的个体差异， 它的建立不是基于现有的认知心理学，无法代表真正的认知过程。

人机交互界面表示模型 LOTOS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- Language Of Temporal Ordering Specification [2]
- 国际标准形式描述语言，无二义性，适于描述具有并发、交互、反馈和不确定性等特点的并发（concurrent）系统中的行为。
- 开始作为一种描述网络协议的语言，由于交互系统、特别是多通道交互系统有并发系统的特点，因此成为用来描述交互系统的行为模型。

人机交互界面表示模型 LOTOS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- Language Of Temporal Ordering Specification [2]
- 国际标准形式描述语言，无二义性，适于描述具有并发、交互、反馈和不确定性等特点的并发（concurrent）系统中的行为。
- 开始作为一种描述网络协议的语言，由于交互系统、特别是多通道交互系统有并发系统的特点，因此成为用来描述交互系统的行为模型。

人机交互界面表示模型 LOTOS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- Language Of Temporal Ordering Specification [2]
- 国际标准形式描述语言，无二义性，适于描述具有并发、交互、反馈和不确定性等特点的并发（concurrent）系统中的行为。
- 开始作为一种描述网络协议的语言，由于交互系统、特别是多通道交互系统有并发系统的特点，因此成为用来描述交互系统的行为模型。

人机交互界面表示模型 LOTOS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 系统的外部可见行为可以看作是由一个有时序关系的交互序列组成。
- 系统由一系列进程组成，进程同环境之间通过称为“关口”（gates）的交互点进行交互。
- 两个以上的进程在执行同一个外部可见的行为时会发生交互操作，进行数据交换、信息传递、协调同步等操作。
- 进程行为用“行为表达式”来描述，复杂的行为由简单的行为表达式通过表示时序关系的LOTOS算符组合而成。
- 在将LOTOS思想用于人机交互的行为模型时，用进程之间的约束关系来描述交互子任务之间的关系。

人机交互界面表示模型 LOTOS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 系统的外部可见行为可以看作是由一个有时序关系的交互序列组成。
- 系统由一系列进程组成，进程同环境之间通过称为“关口”（gates）的交互点进行交互。
- 两个以上的进程在执行同一个外部可见的行为时会发生交互操作，进行数据交换、信息传递、协调同步等操作。
- 进程行为用“行为表达式”来描述，复杂的行为由简单的行为表达式通过表示时序关系的LOTOS算符组合而成。
- 在将LOTOS思想用于人机交互的行为模型时，用进程之间的约束关系来描述交互子任务之间的关系。

人机交互界面表示模型 LOTOS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 系统的外部可见行为可以看作是由一个有时序关系的交互序列组成。
- 系统由一系列进程组成，进程同环境之间通过称为“关口”（gates）的交互点进行交互。
- 两个以上的进程在执行同一个外部可见的行为时会发生交互操作，进行数据交换、信息传递、协调同步等操作。
- 进程行为用“行为表达式”来描述，复杂的行为由简单的行为表达式通过表示时序关系的LOTOS算符组合而成。
- 在将LOTOS思想用于人机交互的行为模型时，用进程之间的约束关系来描述交互子任务之间的关系。

人机交互界面表示模型 LOTOS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 系统的外部可见行为可以看作是由一个有时序关系的交互序列组成。
- 系统由一系列进程组成，进程同环境之间通过称为“关口”（gates）的交互点进行交互。
- 两个以上的进程在执行同一个外部可见的行为时会发生交互操作，进行数据交换、信息传递、协调同步等操作。
- 进程行为用“行为表达式”来描述，复杂的行为由简单的行为表达式通过表示时序关系的LOTOS算符组合而成。
- 在将LOTOS思想用于人机交互的行为模型时，用进程之间的约束关系来描述交互子任务之间的关系。

人机交互界面表示模型 LOTOS

人机交互技术

施展

第六讲

人机交互界面表示模型

型

界面描述语言

窗口系统

用户界面管理系统

小结

- 系统的外部可见行为可以看作是由一个有时序关系的交互序列组成。
- 系统由一系列进程组成，进程同环境之间通过称为“关口”（gates）的交互点进行交互。
- 两个以上的进程在执行同一个外部可见的行为时会发生交互操作，进行数据交换、信息传递、协调同步等操作。
- 进程行为用“行为表达式”来描述，复杂的行为由简单的行为表达式通过表示时序关系的LOTOS算符组合而成。
- 在将LOTOS思想用于人机交互的行为模型时，用进程之间的约束关系来描述交互子任务之间的关系。

人机交互界面表示模型 LOTOS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ LOTOS算符

$T1 \parallel T2$	交替 Interleaving
$T1 \sqcap T2$	选择 Choice
$T1 \mid [a_1, \dots, a_n] \mid T2$	同步 Synchronization
$T1 \lceil T2$	禁止 Deactivation
$T1 \gg T2$	允许 Enabling

人机交互界面表示模型 LOTOS

人机交互技术

施展

第六讲

人机交互界面表示模型

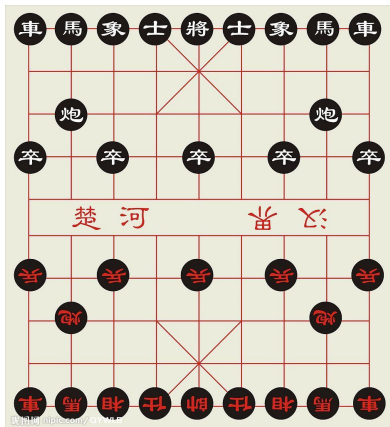
界面描述语言

窗口系统

用户界面管理系统

小结

■ 例：中国象棋



人机交互界面表示模型 LOTOS

人机交互技术

施展

第六讲

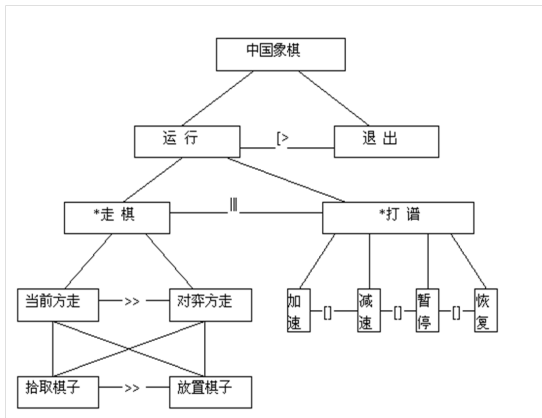
人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结



中国象棋的LOTOS任务分解实例

LOTOS与GOMS的结合

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- LOTOS模型很好的描述了任务之间的时序约束关系，这些时序约束关系能更好的描述GOMS中子目标之间的关系。
- 用GOMS模型描述任务的分解过程，而用LOTOS给出子任务之间的约束关系，这样就可以增加两种表示模型的代表能力。

LOTOS与GOMS的结合

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- LOTOS模型很好的描述了任务之间的时序约束关系，这些时序约束关系能更好的描述GOMS中子目标之间的关系。
- 用GOMS模型描述任务的分解过程，而用LOTOS给出子任务之间的约束关系，这样就可以增加两种表示模型的代表能力。

LOTOS与GOMS的结合

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

GOAL:中国象棋

[>:

GOAL:运行

|||:

*GOAL:走棋

ACTION:自动记录棋谱

>>:

GOAL:当前方走

>>:

OPRATOR:拾取棋子

OPRATOR:放置棋子

GOAL:对弈方走

>>

OPRATOR:拾取棋子

OPRATOR:放置棋子

*GOAL:打谱

[]:

OPRATOR:加速

OPRATOR:减速

OPRATOR:暂停

OPRATOR:恢复

GOAL:退出

LOTOS与GOMS的结合

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- LOTOS与GOMS结合，可以清楚地了解整个目标层次及各目标之间的约束关系。但与GOMS同样存在无法描述目标异常结束的缺陷，同时当任务进行选择时用什么规则进行选择并未涉及。
- LOTOS最大的优越性在于可以构造一套现成的自动化工具，利用这些工具，可自动进行错误检测，但它过于形式化的记法比较晦涩难懂。
- GOMS和LOTOS的结合可以很好地描述人机交互的较高级的任务，对于原子任务的形式化描述，上述模型并没有给出一个比较清晰的描述。
- UAN模型主要用于原子目标的描述。

LOTOS与GOMS的结合

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- LOTOS与GOMS结合，可以清楚地了解整个目标层次及各目标之间的约束关系。但与GOMS同样存在无法描述目标异常结束的缺陷，同时当任务进行选择时用什么规则进行选择并未涉及。
- LOTOS最大的优越性在于可以构造一套现成的自动化工具，利用这些工具，可自动进行错误检测，但它过于形式化的记法比较晦涩难懂。
- GOMS和LOTOS的结合可以很好地描述人机交互的较高级的任务，对于原子任务的形式化描述，上述模型并没有给出一个比较清晰的描述。
- UAN模型主要用于原子目标的描述。

LOTOS与GOMS的结合

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- LOTOS与GOMS结合，可以清楚地了解整个目标层次及各目标之间的约束关系。但与GOMS同样存在无法描述目标异常结束的缺陷，同时当任务进行选择时用什么规则进行选择并未涉及。
- LOTOS最大的优越性在于可以构造一套现成的自动化工具，利用这些工具，可自动进行错误检测，但它过于形式化的记法比较晦涩难懂。
- GOMS和LOTOS的结合可以很好地描述人机交互的较高级的任务，对于原子任务的形式化描述，上述模型并没有给出一个比较清晰的描述。
- UAN模型主要用于原子目标的描述。

LOTOS与GOMS的结合

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- LOTOS与GOMS结合，可以清楚地了解整个目标层次及各目标之间的约束关系。但与GOMS同样存在无法描述目标异常结束的缺陷，同时当任务进行选择时用什么规则进行选择并未涉及。
- LOTOS最大的优越性在于可以构造一套现成的自动化工具，利用这些工具，可自动进行错误检测，但它过于形式化的记法比较晦涩难懂。
- GOMS和LOTOS的结合可以很好地描述人机交互的较高级的任务，对于原子任务的形式化描述，上述模型并没有给出一个比较清晰的描述。
- UAN模型主要用于原子目标的描述。

人机交互界面表示模型 UAN

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ User Action Notation [3]

- UAN是一种简单的符号语言，主要描述用户的行为序列以及在执行任务时所用的界面物理对象。
- 尽管UAN属于一种行为模型，但作为一种任务描述语言，它又涉及一定程度的系统行为的描述，因而它兼有行为模型和结构模型的一些特点。
- ...

人机交互界面表示模型 UAN

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- User Action Notation [3]
- UAN是一种简单的符号语言，主要描述用户的行为序列以及在执行任务时所用的界面物理对象。
- 尽管UAN属于一种行为模型，但作为一种任务描述语言，它又涉及一定程度的系统行为的描述，因而它兼有行为模型和结构模型的一些特点。
- ...

人机交互界面表示模型 UAN

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- User Action Notation [3]
- UAN是一种简单的符号语言，主要描述用户的行为序列以及在执行任务时所用的界面物理对象。
- 尽管UAN属于一种行为模型，但作为一种任务描述语言，它又涉及一定程度的系统行为的描述，因而它兼有行为模型和结构模型的一些特点。

■ ...

人机交互界面表示模型 UAN

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- User Action Notation [3]
- UAN是一种简单的符号语言，主要描述用户的行为序列以及在执行任务时所用的界面物理对象。
- 尽管UAN属于一种行为模型，但作为一种任务描述语言，它又涉及一定程度的系统行为的描述，因而它兼有行为模型和结构模型的一些特点。
- ...

人机交互界面表示模型 结构模型

人机交互技术

施展

第六讲

人机交互界面表示模型

型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 产生式规则 Production Rule

- 上下文无关文法，将人机交互对话看作是一种语言，运用基于语法的方法来描述交互对话。
- 产生式集合定义了用户与计算机交互所运用的语言。

■ 状态转换网络 State Transition Network

- 定义一个具有一定数量状态的转换机，称之为有限状态机-Finite State Machine(FSM)。
- FSM从外部世界中接收到事件，并能使FSM从一个状态转换到另一个状态。
- 两种最基本的状态转换网络，状态转换网络 (State Diagrams) 和扩展状态转换网络(State Charts)，后者是前者的一个扩展。

人机交互界面表示模型 行为模型和结构模型的转换

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 整体框架
- 转换算法

人机交互界面表示模型 表现模型

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 逻辑组织结构
- 面板内部时间分发及响应方式
- 面板间关系

人机交互界面表示模型 表现模型

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 表现模型(PM)描述了用户界面的表现形式，由层次性的交互对象组成。
 - 交互对象一般由抽象交互对象(AIO - Abstract Interactive Object)和具体交互对象(CIO - Concrete Interactive Object)组成。
- 管理信息系统的交互界面：填表界面
 - 界面元素：界面元素属性，对几何对象、内容对象、绘制对象的描述
 - 面板：界面元素的模型定义+界面元素的列表和布局的定义
 - XML描述

人机交互界面表示模型 表现模型

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 表现模型(PM)描述了用户界面的表现形式，由层次性的交互对象组成。
 - 交互对象一般由抽象交互对象(AIO - Abstract Interactive Object)和具体交互对象(CIO - Concrete Interactive Object)组成。
- 管理信息系统的交互界面：填表界面
 - 界面元素：界面元素属性，对几何对象、内容对象、绘制对象的描述
 - 面板：界面元素的模型定义+界面元素的列表和布局的定义
 - XML描述

人机交互界面表示模型 表现模型

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 体验 WinSpy
- 数据结构
- 属性及响应

人机交互界面表示模型 表现模型

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 体验 WinSpy
- 数据结构
- 属性及响应

人机交互界面表示模型 表现模型

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 体验 WinSpy
- 数据结构
- 属性及响应

人机交互界面表示模型 表现模型

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 面板间的关系

- 并列关系
- 嵌套关系
- 依赖关系 依赖面板、依赖事件

■ 窗口界面分类

■ 模态窗口

■ 非模态窗口

■ 对话框

■ 滚动窗口

■ 多文档窗口

人机交互界面表示模型 表现模型

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 面板间的关系

■ 并列关系

■ 嵌套关系

■ 依赖关系 依赖面板、依赖事件

人机交互界面表示模型 表现模型

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 面板间的关系

■ 并列关系

■ 嵌套关系

■ 依赖关系 依赖面板、依赖事件

人机交互界面表示模型 表现模型

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 面板间的关系
 - 并列关系
 - 嵌套关系
 - 依赖关系 依赖面板、依赖事件

■ 面板界面分类

人机交互界面表示模型 表现模型

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 面板间的关系
 - 并列关系
 - 嵌套关系
 - 依赖关系 依赖面板、依赖事件

■ 面板界面分类

人机交互界面表示模型 表现模型

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 面板间的关系
 - 并列关系
 - 嵌套关系
 - 依赖关系 依赖面板、依赖事件
- 面板界面分类
 - 自由面板 (FreePanel)
 - 面板面板 (PanelPanel)
 - 原子面板 (ComponentPanel)

人机交互界面表示模型 表现模型

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 面板间的关系
 - 并列关系
 - 嵌套关系
 - 依赖关系 依赖面板、依赖事件
- 面板界面分类
 - 自由面板 (FreePanel)
 - 面板面板 (PanelPanel)
 - 原子面板 (ComponentPanel)

人机交互界面表示模型 表现模型

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 面板间的关系

- 并列关系
- 嵌套关系
- 依赖关系 依赖面板、依赖事件

■ 面板界面分类

- 自由面板 (FreePanel)
- 面板面板 (PanelPanel)
- 原子面板 (ComponentPanel)

体验 Qt Designer + Qt Sp

人机交互界面表示模型 表现模型

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 面板间的关系

- 并列关系
- 嵌套关系
- 依赖关系 依赖面板、依赖事件

■ 面板界面分类

- 自由面板 (FreePanel)
- 面板面板 (PanelPanel)
- 原子面板 (ComponentPanel)

体验 Qt Designer + Qt Sp

人机交互界面表示模型 表现模型

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 面板间的关系
 - 并列关系
 - 嵌套关系
 - 依赖关系 依赖面板、依赖事件
- 面板界面分类
 - 自由面板 (FreePanel)
 - 面板面板 (PanelPanel)
 - 原子面板 (ComponentPanel)
- 体验 QT Designer + XML Spy

界面描述语言

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 命令式语言

- 由编程人员明确指定如何执行任务
- C++, Java, Python, 各类过程性语言 ...

■ 陈述式语言

- 由编程人员说明任务内容和目的
 - XML, RDF, LaTeX, 各类标记语言 ...
- 陈述式语言更为抽象、通用、适合于界面描述。

界面描述语言

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 命令式语言

- 由编程人员明确指定如何执行任务
- C++, Java, Python, 各类过程性语言 ...

■ 陈述式语言

- 由编程人员说明任务内容和目的
- XML, RDF, LaTeX, 各类标记语言 ...

■ 陈述式语言更为抽象、通用、适合于界面描述。

界面描述语言

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 命令式语言
 - 由编程人员明确指定如何执行任务
 - C++, Java, Python, 各类过程性语言 ...
- 陈述式语言
 - 由编程人员说明任务内容和目的
 - XML, RDF, LaTeX, 各类标记语言 ...
- 陈述式语言更为抽象、通用、适合于界面描述。

常见陈述性界面描述语言

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 用户界面标记语言 UIML

■ User Interface Markup Language

■ 由结构 (structure)、样式 (style)、内容 (content)、行为 (behavior)四个方面来描述 [1]

■ 由 Organization for the Advancement of Structured Information Standards (OASIS) 标准化

■ 扩展界面标记语言 XI ML

■ 由组件 (Components)、关系 (Relations)和属性 (Attributes)三部分构成

■ 组件：定义了任务、域、用户、表现和对话五类

■ XML用户界面语言 XUL

■ 提供创建现代图形界面大多数元素的能力。

常见陈述性界面描述语言

人机交互技术

施展

第六讲

人机交互界面表示模型

型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 用户界面标记语言 UIML

■ User Interface Markup Language

■ 由结构 (structure)、样式 (style)、内容 (content)、行为 (behavior) 四个方面来描述 [1]

■ 由 Organization for the Advancement of Structured Information Standards (OASIS) 标准化

■ 扩展界面标记语言 XIML

■ 由组件 (Components)、关系 (Relations) 和属性 (Attributes) 三部分构成

■ 组件：定义了任务、域、用户、表现和对话五类

■ XML 用户界面语言 XUL

■ 提供创建现代图形界面大多数元素的能力。

常见陈述性界面描述语言

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 用户界面标记语言 UIML
 - User Interface Markup Language
 - 由结构 (structure)、样式 (style)、内容 (content)、行为 (behavior)四个方面来描述 [1]
 - 由 Organization for the Advancement of Structured Information Standards (OASIS) 标准化
- 扩展界面标记语言 XIML
 - 由组件 (Components)、关系 (Relations)和属性 (Attributes)三部分构成
 - 组件：定义了任务、域、用户、表现和对话五类
- XML用户界面语言 XUL
 - 提供创建现代图形界面大多数元素的能力。

HTML/CSS

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

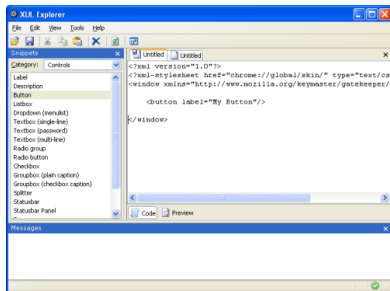
- HTML 超文本标签语言
- XHTML 更严谨HTML版本
- HTML 5 下一代的HTML
- CSS 层叠样式表 Cascading Style Sheets
- 体验 Chrome

- HTML 超文本标签语言
- XHTML 更严谨HTML版本
- HTML 5 下一代的HTML
- CSS 层叠样式表 Cascading Style Sheets
- 体验 Chrome

■ XML User Interface Language

- 支持Mozilla系列的应用程序(如Mozilla Firefox和Mozilla Thunderbird)而开发的用户界面标记语言。
- 重用了许多现有的标准和技术 包括CSS、JavaScript、DTD和RDF等。
- 主要好处在于提供了一套简易和跨平台的widget定义。

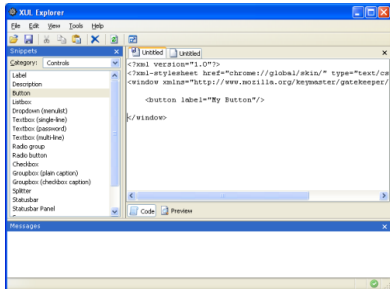
■ 体验 QT Designer



■ XML User Interface Language

- 支持Mozilla系列的应用程序(如Mozilla Firefox和Mozilla Thunderbird)而开发的用户界面标记语言。
- 重用了许多现有的标准和技术 包括CSS、JavaScript、DTD和RDF等。
- 主要好处在于提供了一套简易和跨平台的widget定义。

■ 体验 QT Designer



窗口系统

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 对交互应用系统实现三个层次上的支持
 - 窗口系统对交互应用系统实现的支持
 - 软件开发环境对交互应用系统实现的支持
 - 用户界面管理系统UIMS

窗口系统

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 对交互应用系统实现三个层次上的支持
 - 窗口系统对交互应用系统实现的支持
 - 软件开发环境对交互应用系统实现的支持
 - 用户界面管理系统UIMS

窗口系统

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 对交互应用系统实现三个层次上的支持
 - 窗口系统对交互应用系统实现的支持
 - 软件开发环境对交互应用系统实现的支持
 - 用户界面管理系统UIMS

窗口系统

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 对交互应用系统实现三个层次上的支持
 - 窗口系统对交互应用系统实现的支持
 - 软件开发环境对交互应用系统实现的支持
 - 用户界面管理系统UIMS

窗口系统结构

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 三种不同结构

■ 在各个应用程序内部实现和管理多任务

- 移植起来不方便
- 应用程序处理复杂

■ 在操作系统核心集中处理多任务管理

- 过分的依赖操作系统
- 移植起来不方便

■ 多任务的管理可由独立的管理程序进行管理

- 应用程序通过调用该管理程序提供的接口来实现对多任务的管理和设备的独立性操作
- 最容易移植的

窗口系统结构

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 三种不同结构

■ 在各个应用程序内部实现和管理多任务

- 移植起来不方便
- 应用程序处理复杂

■ 在操作系统核心集中处理多任务管理

- 过分的依赖操作系统
- 移植起来不方便

■ 多任务的管理可由独立的管理程序进行管理

- 应用程序通过调用该管理程序提供的接口来实现对多任务的管理和设备的独立性操作
- 最容易移植的

窗口系统结构

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 三种不同结构

■ 在各个应用程序内部实现和管理多任务

- 移植起来不方便
- 应用程序处理复杂

■ 在操作系统核心集中处理多任务管理

- 过分的依赖操作系统
- 移植起来不方便

■ 多任务的管理可由独立的管理程序进行管理

- 应用程序通过调用该管理程序提供的接口来实现对多任务的管理和设备的独立性操作
- 最容易移植的

独立管理程序的客户/服务器结构

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

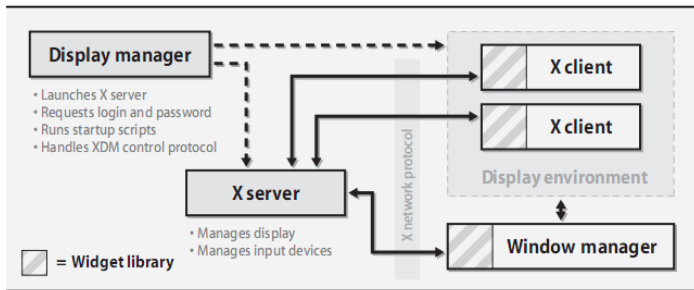
窗口系统

用户界面管理系统

小结

■ 以X Window为例

The X client/server model



管理程序组成

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 窗口系统由在服务器端运行的三部分程序组成

■ 资源管理器

- 整个窗口系统的核心
- 负责多任务的管理，并通过设备驱动程序来管理外部设备。

■ 设备驱动程序

- 负责外部设备的驱动，接受输入设备的输入。
- 将输入数据转换成统一的格式，通过设备驱动程序实现设备的独立性。

■ 抽象终端

- 负责和客户应用程序的接口
- 对每个应用程序由窗口管理程序为其分配一个抽象终端。

管理程序组成

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 窗口系统由在服务器端运行的三部分程序组成

■ 资源管理器

- 整个窗口系统的核心
- 负责多任务的管理，并通过设备驱动程序来管理外部设备。

■ 设备驱动程序

- 负责外部设备的驱动，接受输入设备的输入。
- 将输入数据转换成统一的格式，通过设备驱动程序实现设备的独立性。

■ 抽象终端

- 负责和客户应用程序的接口
- 对每个应用程序由窗口管理程序为其分配一个抽象终端。

管理程序组成

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 窗口系统由在服务器端运行的三部分程序组成
 - 资源管理器
 - 整个窗口系统的核心
 - 负责多任务的管理，并通过设备驱动程序来管理外部设备。
 - 设备驱动程序
 - 负责外部设备的驱动，接受输入设备的输入。
 - 将输入数据转换成统一的格式，通过设备驱动程序实现设备的独立性。
 - 抽象终端
 - 负责和客户应用程序的接口
 - 对每个应用程序由窗口管理程序为其分配一个抽象终端。

管理程序组成

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 窗口系统由在服务器端运行的三部分程序组成
 - 资源管理器
 - 整个窗口系统的核心
 - 负责多任务的管理，并通过设备驱动程序来管理外部设备。
 - 设备驱动程序
 - 负责外部设备的驱动，接受输入设备的输入。
 - 将输入数据转换成统一的格式，通过设备驱动程序实现设备的独立性。
 - 抽象终端
 - 负责和客户应用程序的接口
 - 对每个应用程序由窗口管理程序为其分配一个抽象终端。

交互事件处理

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 主要有两种形式

■ 应用程序内部事件处理循环

- 服务器把用户的输入作为事件送给客户应用程序
- 客户应用程序对传给它的的所有的事件都做出响应，不同的事件采取不同的处理
- 早期的基于窗口系统的开发往往采用这种方式

■ 事件注册方式

- 事件处理中心负责事件的处理；
- 应用程序登记处理的事件；
- 事件处理中心接收事件，把事件和控制转向该事件注册的回应过程；
- 处理完后，回应过程把控制返还给事件处理中心。

■ 事件注册方式优点

- 应用程序不需要设计事件处理循环
- 事件处理中心处理事件的效率相对比较高

交互事件处理

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 主要有两种形式

■ 应用程序内部事件处理循环

- 服务器把用户的输入作为事件送给客户应用程序
- 客户应用程序对传给它的的所有的事件都做出响应，不同的事件采取不同的处理
- 早期的基于窗口系统的开发往往采用这种方式

■ 事件注册方式

- 事件处理中心负责事件的处理；
- 应用程序登记处理的事件；
- 事件处理中心接收事件，把事件和控制转向该事件注册的回应过程；
- 处理完后，回应过程把控制返还给事件处理中心。

■ 事件注册方式优点

- 应用程序不需要设计事件处理循环
- 事件处理中心处理事件的效率相对比较高

交互事件处理

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 主要有两种形式

■ 应用程序内部事件处理循环

- 服务器把用户的输入作为事件送给客户应用程序
- 客户应用程序对传给它的的所有的事件都做出响应，不同的事件采取不同的处理
- 早期的基于窗口系统的开发往往采用这种方式

■ 事件注册方式

- 事件处理中心负责事件的处理；
- 应用程序登记处理的事件；
- 事件处理中心接收事件，把事件和控制转向该事件注册的回应过程；
- 处理完后，回应过程把控制返还给事件处理中心。

■ 事件注册方式优点

- 应用程序不需要设计事件处理循环
- 事件处理中心处理事件的效率相对比较高

交互事件处理

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

■ 主要有两种形式

■ 应用程序内部事件处理循环

- 服务器把用户的输入作为事件送给客户应用程序
- 客户应用程序对传给它的的所有的事件都做出响应，不同的事件采取不同的处理
- 早期的基于窗口系统的开发往往采用这种方式

■ 事件注册方式

- 事件处理中心负责事件的处理；
- 应用程序登记处理的事件；
- 事件处理中心接收事件，把事件和控制转向该事件注册的回应过程；
- 处理完后，回应过程把控制返还给事件处理中心。

■ 事件注册方式优点

- 应用程序不需要设计事件处理循环
- 事件处理中心处理事件的效率相对比较高

用户界面管理系统

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 一个支持交互系统开发的UIMS的概念结构
- 该结构把应用程序的语义与表现分开
- 保留应用程序和表示形式之间的内在关系
- 支持运行的交互系统的管理、实现和评估的技术

用户界面管理系统

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 一个支持交互系统开发的UIMS的概念结构
- 该结构把应用程序的语义与表现分开
- 保留应用程序和表示形式之间的内在关系
- 支持运行的交互系统的管理、实现和评估的技术

用户界面管理系统

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 一个支持交互系统开发的UIMS的概念结构
- 该结构把应用程序的语义与表现分开
- 保留应用程序和表示形式之间的内在关系
- 支持运行的交互系统的管理、实现和评估的技术

用户界面管理系统

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 一个支持交互系统开发的UIMS的概念结构
- 该结构把应用程序的语义与表现分开
- 保留应用程序和表示形式之间的内在关系
- 支持运行的交互系统的管理、实现和评估的技术

UIMS的表示方法

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结



■ 表现层表示方法

- 用户输入输出信息、图形表示的处理
- 适应多媒体的需要、智能人机界面规格说明

■ 对话控制的表示方法

■ 基于语言的表示方法

菜单网络，状态转换网络…

■ 基于图形的表示方法

用户使用鼠标器直接将对象放到屏幕上来定义界面

■ 基于应用语义过程的表示方法

■ 应用层的表示方法

- 有关应用数据结构、子程序、及对用户的限制
- 排除可能引起语义错误的操作
- 避免对应用程序的破坏

UIMS的表示方法

人机交互技术

施展

第六讲

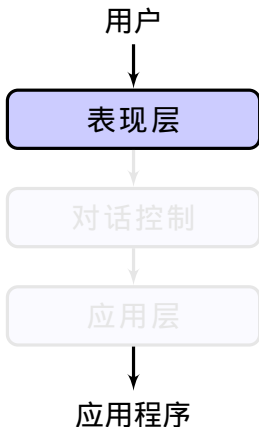
人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结



■ 表现层表示方法

- 用户输入输出信息、图形表示的处理
- 适应多媒体的需要、智能人机界面规格说明

■ 对话控制的表示方法

■ 基于语言的表示方法

菜单网络，状态转换网络…

■ 基于图形的表示方法

用户使用鼠标器直接将对象放到屏幕上来定义界面

■ 基于应用语义过程的表示方法

■ 应用层的表示方法

- 有关应用数据结构、子程序、及对用户的限制
- 排除可能引起语义错误的操作
- 避免对应用程序的破坏

UIMS的表示方法

人机交互技术

施展

第六讲

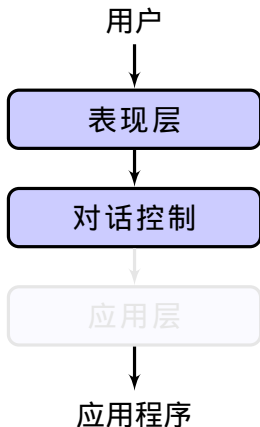
人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结



■ 表现层表示方法

- 用户输入输出信息、图形表示的处理
- 适应多媒体的需要、智能人机界面规格说明

■ 对话控制的表示方法

■ 基于语言的表示方法

菜单网络，状态转换网络...

■ 基于图形的表示方法

用户使用鼠标器直接将对象放到屏幕上来定义界面

■ 基于应用语义过程的表示方法

■ 应用层的表示方法

- 有关应用数据结构、子程序、及对用户的限制
- 排除可能引起语义错误的操作
- 避免对应用程序的破坏

UIMS的表示方法

人机交互技术

施展

第六讲

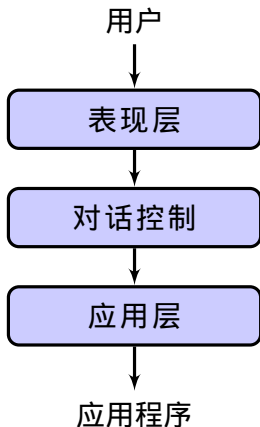
人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结



■ 表现层表示方法

- 用户输入输出信息、图形表示的处理
- 适应多媒体的需要、智能人机界面规格说明

■ 对话控制的表示方法

■ 基于语言的表示方法

菜单网络，状态转换网络...

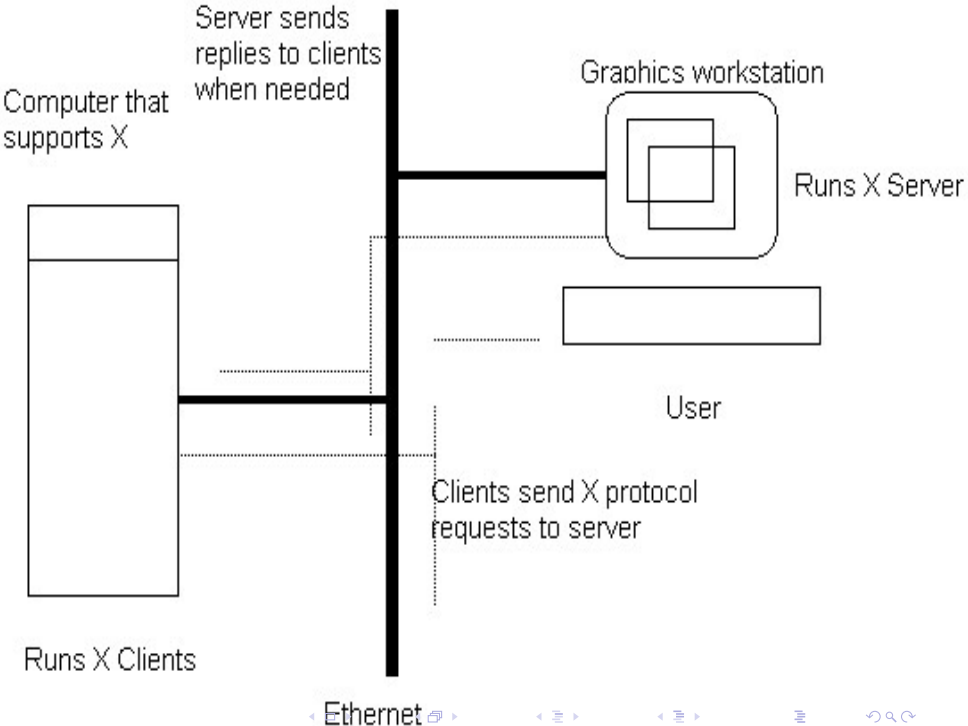
■ 基于图形的表示方法

用户使用鼠标器直接将对象放到屏幕上来定义界面

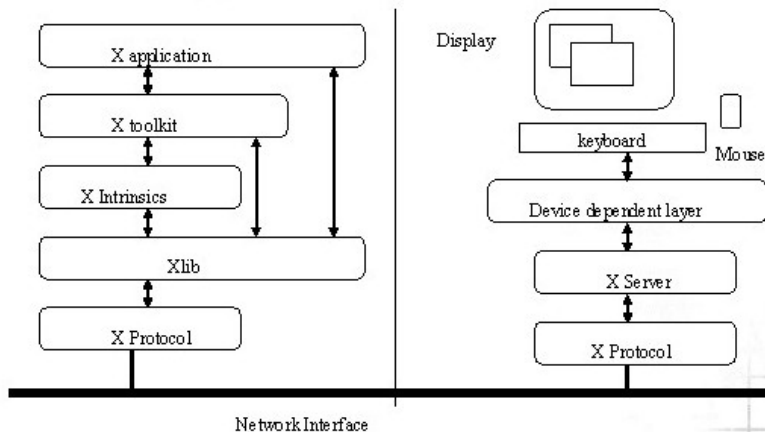
■ 基于应用语义过程的表示方法

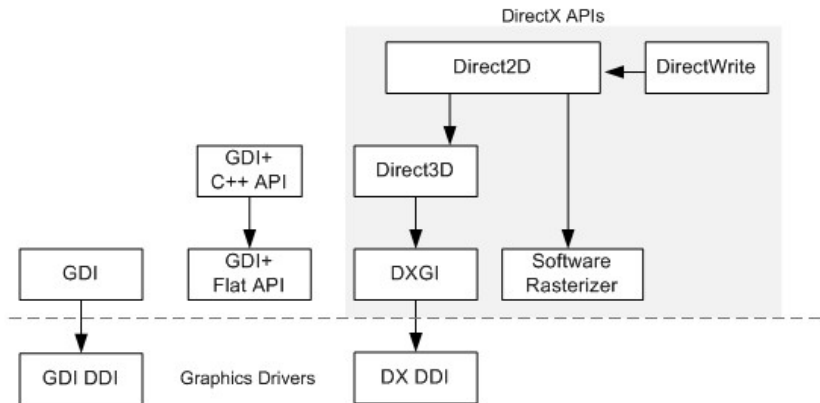
■ 应用层的表示方法

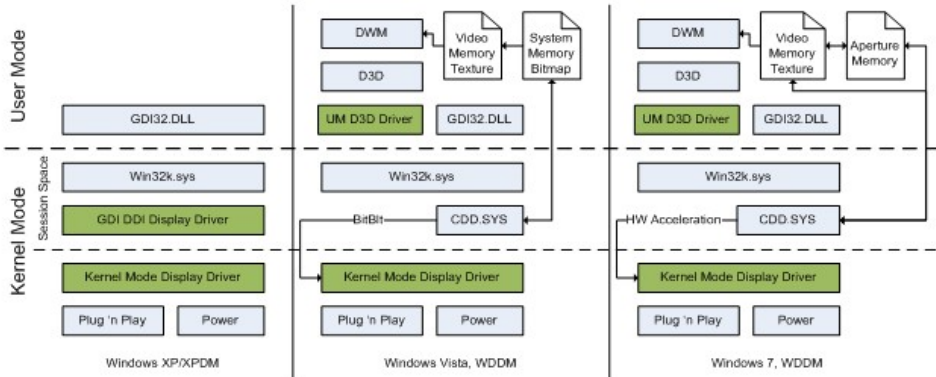
- 有关应用数据结构、子程序、及对用户的限制
- 排除可能引起语义错误的操作
- 避免对应用程序的破坏



X Window application Structure







Windows Presentation Foundation

人机交互技术

施展

第六讲

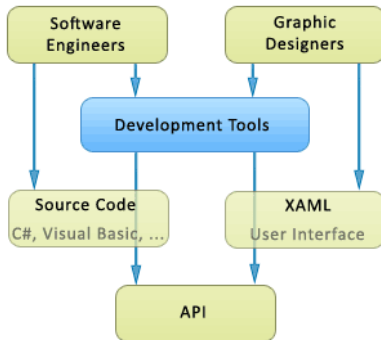
人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结



其底层使用 Extensible Application Markup Language (XAML)

SELECTORS		
#id, tag, .class, *	E[@attr]	
elm1, elm2, elmN	E[@attr=val]	
ancestor descendant	E[@attr^=val] (begins)	
parent > child	E[@attr\$=val] (ends)	
parent/child	E[@attr*=val] (contains)	
prev + next	E[@attr=val][@attr=val] (both)	
prev ~ siblings		
	:nth-child(index)	
:first	:first-child	
:last	:last-child	
:not(selector)	:only-child	:input
:even		:text
:odd	:enabled	:password
:eq(index)	:disabled	:radio
:gt(index)	:checked	:checkbox
:lt(index)	:selected	:submit
		:image
:contains(text)	:hidden	:reset
:empty	:visible	:button
:has(selector)	:header	:file
:parent	:animated	:hidden

CSS		ATTRIBUTES
.css(name, value)	.attr(name)	.attr(properties)
.css(properties)	.attr(key, value)	.attr(key, function)
	.removeAttr(name)	
.height(value)		
.width(value)		
.addClass(class)		
.removeClass(class)		
.toggleClass(class)		
.offset()		
HTML		
	.html()	.html(value)
	.text(), .text(value)	.val(value)


USER AGENT	JavaScript	
\$browser, \$browser.version	\$extend(obj1,...objN)	\$map(array, callback)
\$boxModel	\$grep(array, callback, invert)	\$unique(array)
		\$trim(string)
		\$merge(1st, 2nd)

EVENTS		
HANDLERS	ERROR	INTERACTION
.bind(type, data, fn)	.error()	.hover(fnIN, fnOUT)
.one(type, data, fn)	.error(fn)	.toggle(fnIN, fnOUT)
.trigger(type, data)		.blur() .blur(fn)
.triggerHandler(type, data)		.change() .change(fn)
.unbind(type, data)	KEYBOARD	.click() .click(fn)
MOUSE	.keydown()	.dblclick() .dblclick(fn)
.mousedown(fn)	.keydown(fn)	.focus() .focus(fn)
.mousemove(fn)	.keypress()	.select() .select(fn)
.mouseout(fn)	.keypress(fn)	.submit() .submit(fn)
.mouseover(fn)	.keyup()	.unload() .unload(fn)
.mouseup(fn)	.keyup(fn)	
WINDOW		.unblur() .unblur(fn)
.load(fn)	.scroll(fn)	
.resize(fn)	PAGE	
	.ready(fn)	

TRAVERSING		MANIPULATING
FILTER	ACCESS	INSIDE (content)
.hasClass(class)	.each(callback)	.append(c)
.filter(expr)	.size()	.appendTo(c)
.filter(fn)	.length	.prepend(c)
.is(expr)	.get()	.prependTo(c)
.map(callback)	.get(index)	
.not(expr)	.index(subject)	
.slice(start, end)		
FIND (expr)	CHAIN	AROUND
.add(e)	.andSelf()	.wrap(html)
.children(e), .siblings(e)	.end()	.wrap(element)
.contents()		.wrapAll(html)
.find(e)		.wrapAll(element)
.next(e), .nextAll(expr)		.wrapInner(html)
.parent(e), .parents(e)		.wrapInner(element)
.prev(e), .prevAll(e)		

CORE UI EFFECTS	
SHOW / HIDE	SLIDE (speed, callback)
.show()	.slideDown(s, c)
.show(speed, callback)	.slideUp(s, c)
.hide()	.slideToggle(s, c)
.hide(speed, callback)	
.toggle()	
ANIMATE	FADE
.stop()	.fadeIn(speed, callback)
.queue()	.fadeOut(speed, callback)
.queue(callback),	.fadeTo(speed, opacity, callback)
.queue(queue)	
.dequeue()	
.animate(params, duration, easing, callback)	
.animate(params, options)	

AJAX	
Request (url, data, callback)	
\$.ajax(options)	
.load(u, d, c)	
\$.get(u, d, c)	
\$.getJSON(u, d, c)	
\$.getScript(u, c)	
\$.post(u, d, c)	
.loadIfModified(u, d, c)	
Event Handler (callback)	
.ajaxComplete(c)	.ajaxError(c)
.ajaxSend(c)	.ajaxStart(c)
.ajaxStop(c)	.ajaxSuccess(c)
Serialize	
.serialize()	
.serializeArray()	
.ajaxSetup(options)	

 COLORCHARGE http://colorcharge.com	EXTEND	\$();
jQuery 1.2 Cheat-sheet updated: December 23rd, 2007	\$.fn.extend(obj) \$.extend(obj) \$.noConflict()	\$(expression, context), \$(html) \$(elements), \$(callback)



jQuery Base			Basic Effects	Advanced Events
DOM Traversing	Events	Style	show(String)	ready(Function)
find(String)	bind(String,Function)	css(Key,Value)	hide(String)	hover(Function,Function)
end()	unbind(String,Function)	css(Hash)	slideDown(String)	event(Function)
filter(String)	DOM Modification	addClass(String)	slideUp(String)	unevent(Function)
not(String)	set(Name,Value)	removeClass(String)	fadeIn(String)	onebind(Function)
not(DOMElement)	set(Hash)	Effects	fadeOut(String)	
add(String)	html(String)	hide()	center()	
add(Array)	remove()	show()		
add(DOMElement)	append(String)	toggle()		
Iteration	append(DOMElement,...)	\$()		
size()	append(Array)	\$(Expression)		AJAX Plugin
get()	prepend(...)	\$(DOMElement)		load(URI)
get(N)	before(...)	\$(...,Context)		load(URI,Params,Callback)
each(Function)	after(...)			\$.get(URI,Callback)
	wrap(...)			\$.post(URI,Params,Callback)

```
<head>
```

```
<script type="text/javascript" src="jquery.js"></script>
```

Include the jquery file

```
<script type="text/javascript">
```

The "ready event" (Binds a function to be executed whenever the DOM is ready)

```
$(document).ready(function(){
```

This part can be written in an external .js file.

```
$(".button").click(function(){
```

Where do you want to bind the function?

It can be CSS class, ID, Selectors (ie. DIV, H1, A, P, LI...)

```
$("#panel").slideDown("slow");
```

This function will be triggered when an element with class="button" is clicked

```
});
```

What would like to do with #panel?

In this case, slide it down with "slow" speed.

```
});
```

```
</script>
```

Where do you want to apply this function?

In this case, it is the element with id="panel"

```
</head>
```

```
$("#panel")
```

The quotation marks can be either single or double.
ie. (.class") or ('.class')

```
<head>
```

```
<script type="text/javascript" src="jquery.js"></script>
```

Include the jquery file

jQuery

- 是一个 JavaScript 库
- 极大地简化了 JavaScript 编程
- 很容易学习

This part can be written
in an external .js file

```
</script>
```

```
</head>
```

In this case, since it doesn't have a class, you can use the id attribute.

Where do you want to apply this function?
In this case, it is the element with id="panel"

```
$("#panel")
```

The quotation marks can be either single or double.
ie. ("class") or ('.class')

小结

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结

- 理解人机交互界面表示模型
- 了解界面描述语言、窗口系统、用户界面管理系统

参考文献

人机交互技术

施展

第六讲

人机交互界面表示模型

界面描述语言

窗口系统

用户界面管理系统

小结



M. Abrams, C. Phanouriou, A.L. Batongbacal, S.M. Williams, and J.E. Shuster.

Uiml: an appliance-independent xml user interface language.

Computer Networks, 31(11):1695--1708, 1999.



T. Bolognesi and E. Brinksma.

Introduction to the iso specification language lotos.

Computer Networks and ISDN systems, 14(1):25--59, 1987.



H.R. Hartson, A.C. Siochi, and D. Hix.

The uan: A user-oriented representation for direct manipulation interface designs.

ACM Transactions on Information Systems (TOIS), 8(3): 181--203, 1990.