接口和界面设计

Chapter 8

北京航空航天大學
SCHOOL OF SOFTWARE BEIHANG UNIVERSITY 软件学院

内容安排

- 接口/界面设计基础
- 用户界面设计
- 输入设计
- 输出设计

内容安排

- 接口/界面设计基础
- 用户界面设计
- 输入设计
- 输出设计

用户界面设计的主要工作

- 分析阶段，找到所需要的最低限度的边界类
- 设计阶段，针对用户界面，进一步研究用例场景的实现细节，设计最终的用户界面
  - 页面数及布局
  - 页面的迁移
- 设计阶段，针对系统接口，设计单独的接口和实现类来封装接口逻辑
  - 接口设计
  - 实现类的设计

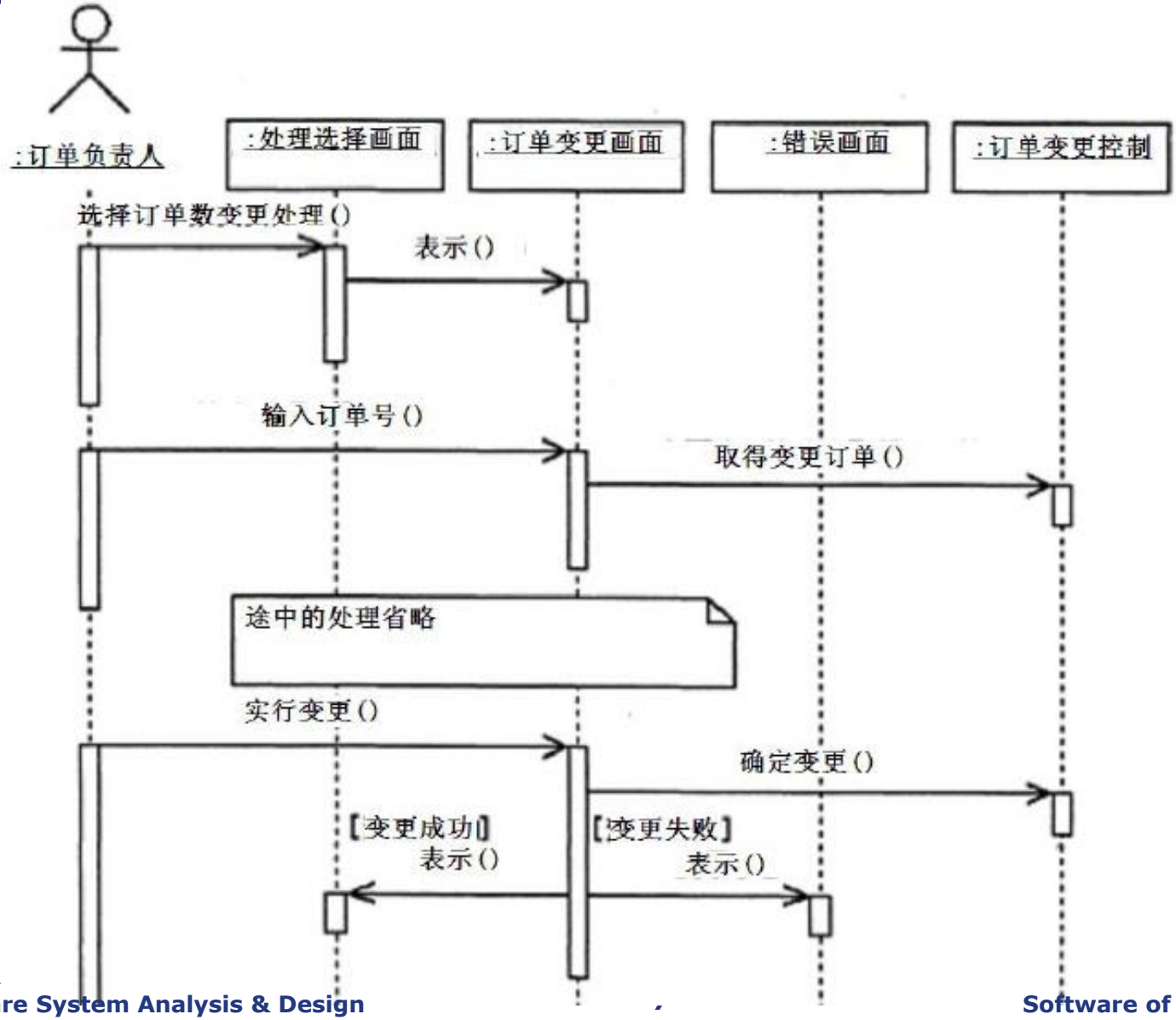
页面数及布局设计

- 用户界面是系统使用者与系统的接触点，其设计质量对系统的整体评价有非常重要的影响
- 页面数
  - 从顺序图中找到参与者进行系统操作时的重点、或者显示结果的重点，考虑所需要的页面数
- 布局设计
  - 基于用户和系统的交互，考虑页面所需内容，进行布局设计
  - 布局要考虑用户的特性进行设计
  - 在早期作出、验证原型，布局设计就能顺利地展开
  - 基于原型，设计符合开发环境和程序语言的布局

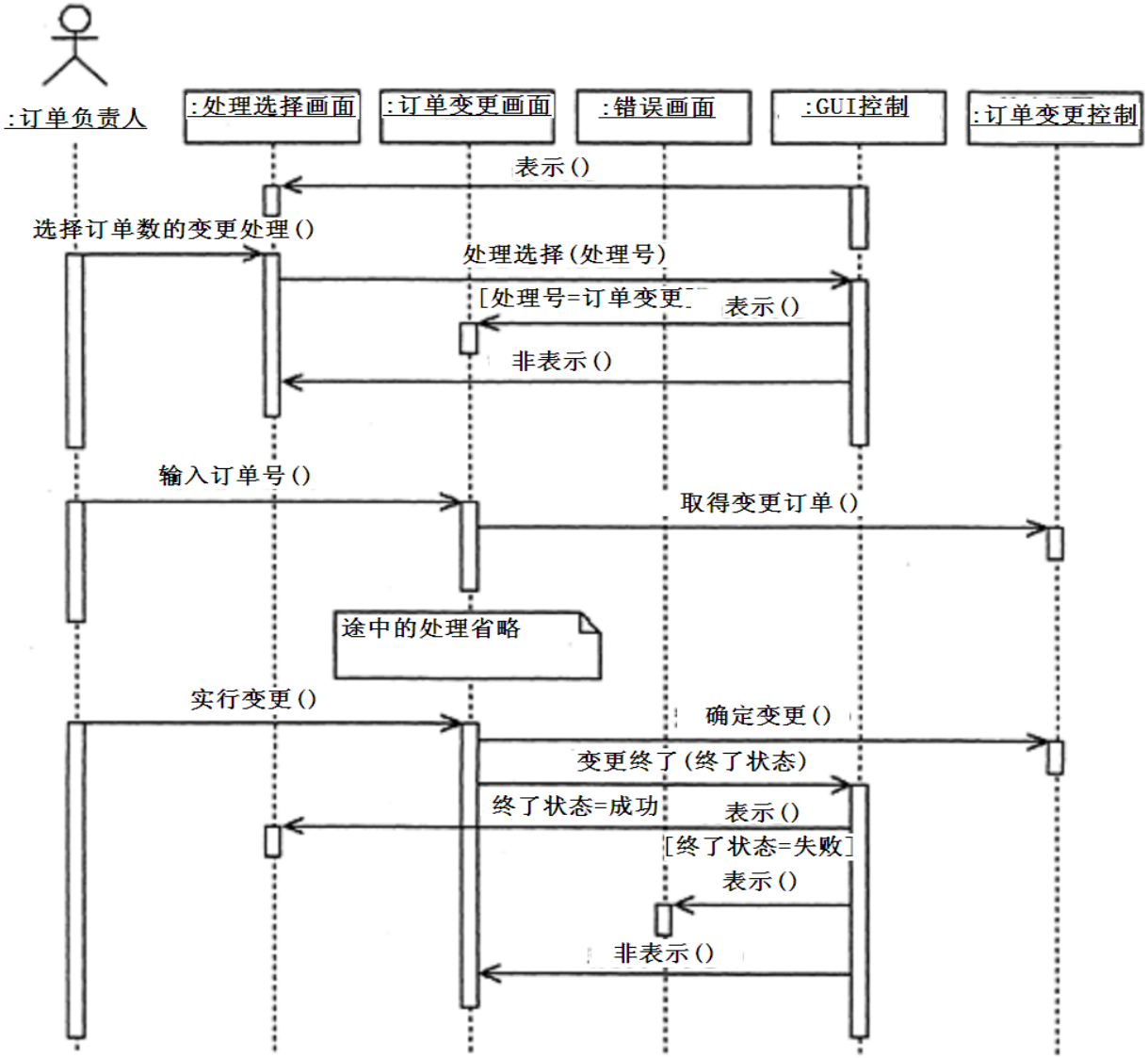
页面迁移

- 当多个页面与1个场景关联时，需要控制页面的迁移
  - 方案1：各个页面知道前后页面，并对应用户的操作显示
    - 前后页面连接较强，对场景变更应对能力很弱
  - 方案2：准备控制页面迁移的专用类(GUI控制类)，对全部的页面进行生成、删除
    - 页面之间的联结弱，对场景变更应对能力很强
    - GUI控制类虽然有控制类的含义，但只有控制页面的功能，不进行应用程序的控制；也被归为边界包

方案1：由页面本身控制迁移



方案2：设计页面迁移控制类



内容安排

- 接口/界面设计基础
- 用户界面设计
- 输入设计
- 输出设计

System User Classifications

- Expert User – an experienced computer user
  - Spends considerable time using specific application programs.
  - Use of a computer is usually considered non-discretionary.
- Novice User – a less experienced computer user
  - Uses computer on a less frequent, or even occasional, basis.
  - Use of a computer may be viewed as discretionary (although this is becoming less and less true).

Commandments of User Interface Design

- Understand your users and their tasks.
- Involve the user in interface design.
- Test the system on actual users.
- Practice iterative design.

Human Engineering Guidelines

- The user should always be aware of what to do next
  - Tell user what the system expects right now.
  - Tell user that data has been entered correctly.
  - Tell user that data has not been entered correctly.
  - Explain reason for a delay in processing.
  - Tell user a task was completed or not completed.
- Format screen so instructions and messages always appear in same general display area.
- Display messages and instructions long enough so user can read them.

Human Engineering Guidelines (continued)

- Use display attributes sparingly.
- Default values should be specified.
- Anticipate errors users might make.
- Users should not be allowed to proceed without correcting an error.
- If user does something that could be catastrophic, the keyboard should be locked to prevent any further input, and an instruction to call the analyst or technical support should be displayed.

Guidelines for dialogue Tone and Terminology

- Dialogue – the overall flow of screens and messages for an application
  - Tone:
    - Use simple, grammatically correct sentences.
    - Don't be funny or cute!
    - Don't be condescending.
  - Terminology
    - Don't use computer jargon.
    - Avoid most abbreviations.
    - Use simple terms.
    - Be consistent in your use of terminology.
    - Carefully phrase instructions—use appropriate action verbs.

User Interface Technology

- Operating Systems and Web Browsers
  - Windows, Macintosh, UNIX, Linux, iOS, Android
  - Growing importance of platform independence
- Display Monitor
  - Regular PC monitors
  - Non-GUI terminals
  - Growing importance of devices such as handhelds
- Keyboards and Pointers
  - Mouse
  - Pens

Graphical User Interfaces Styles and Considerations

- Windows / Pages and frames
- Menu-driven interfaces
  - Pull-down and cascading menus
  - Tear-off and pop-up menus
  - Toolbar and iconic menus
  - Hypertext and hyperlink menus
- Instruction-driven interfaces
  - Language-based syntax
  - Mnemonic syntax
  - Natural language syntax
- Question-answer dialogue

Special Considerations for User Interface Design

- Internal Controls – Authentication and Authorization
  - User ID and Password
  - Privileges assigned to roles
  - Web certificates
- Online Help
  - Growing use of HTML for help systems
  - Tool tips
  - Help wizards
  - Agents – reusable software object that can operate across different applications and networks.

The User Interface Design Process

1. Chart the user interface dialogue.
   - State Transition Diagram– a tool used to depict the sequence and variation of screens that can occur during a user session.
2. Prototype the dialogue and user interface.
3. Obtain user feedback.
   - Exercising (or testing) the user interface
4. If necessary return to step 1 or 2

Contents

- Interface Design Tasks
- User Interface Design
- Input Design
- Output Design

An Input Taxonomy

- Inputs can be classified according to two characteristics
  - How the data is initially captured, entered, and processed
  - The method and technology used to capture and enter the data.

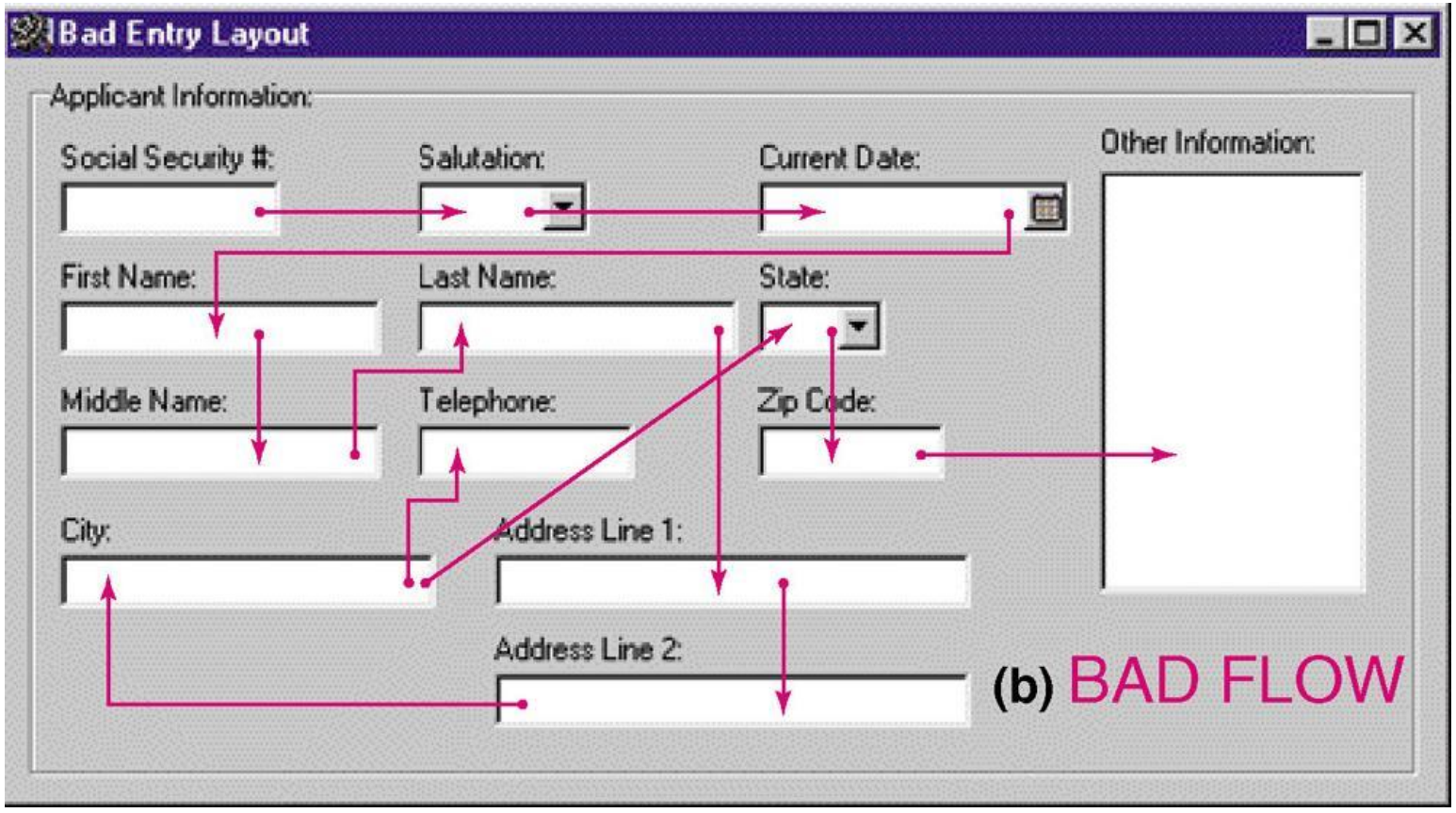
Input Implementation Methods

- Keyboard
- Mouse
- Touch Screen
- Point-of-sale terminals
- Sound and speech
- Automatic data capture
  - Optical mark recognition (OMR)
    - Bar codes
  - Optical character recognition (OCR)
  - Magnetic Ink
  - Electromagnetic transmission
  - Smart cards
  - Biometric

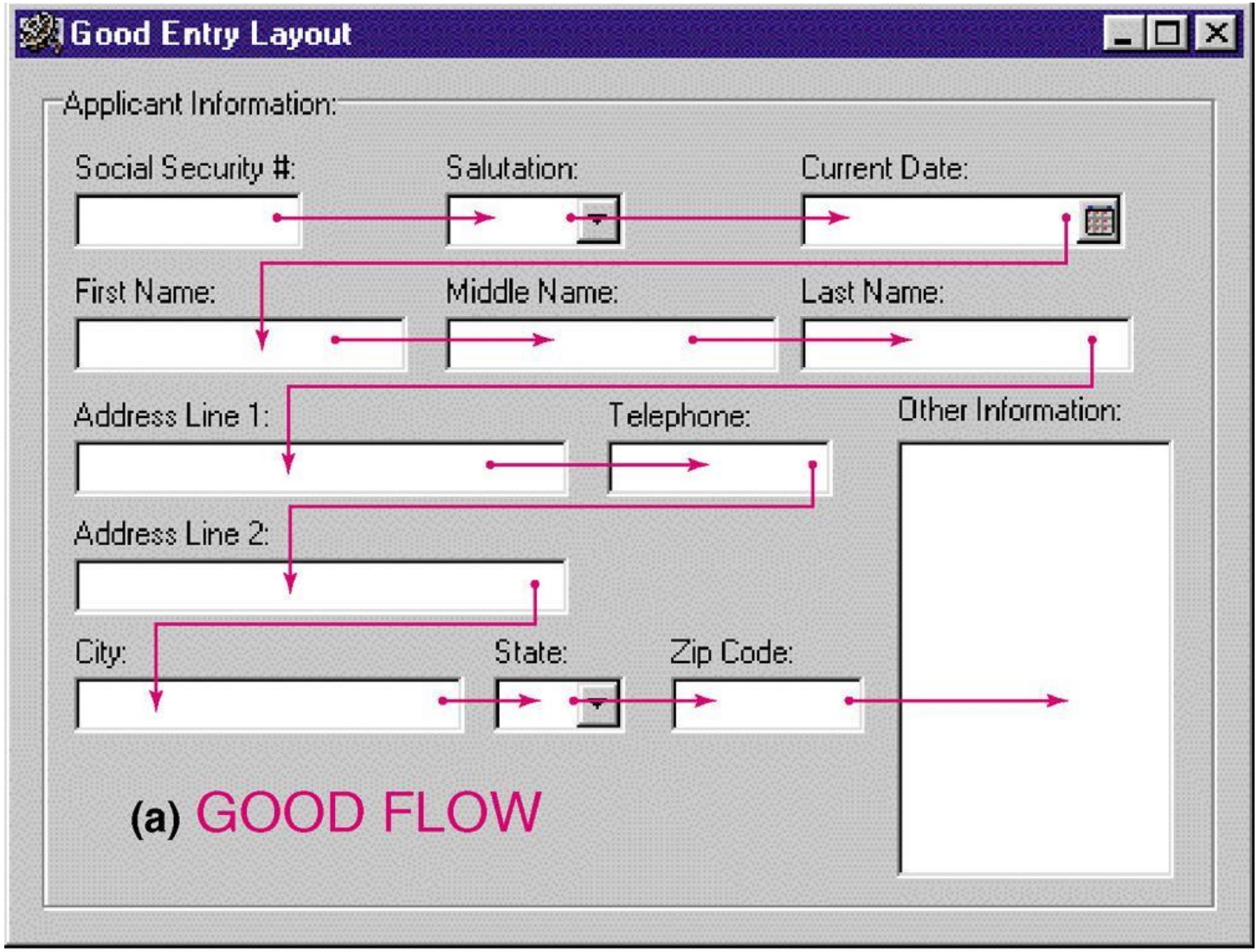
Input Design Guidelines

- Capture only variable data.
  - Not data that can be looked up.
- Do not capture data that can calculated or stored in computer programs as constants.
  - Extended Price, Federal Withholding, etc.
- Use codes for appropriate attributes.

Bad Flow in a Form



Good Flow in a Form



Internal Controls for Inputs

- The number of inputs should be monitored (to minimize risk of lost transactions).
- Validate all data
  - Existence checks
  - Data-type checks
  - Domain checks
  - Combination checks
  - Self-checking digits
  - Format checks

Input Design Process

1. Identify system inputs and review logical requirements.
2. Select appropriate GUI controls.
3. Design, validate and test inputs using some combination of:
   a) Layout tools (e.g., hand sketches, spacing charts, or CASE tools.)
   b) Prototyping tools (e.g., spreadsheet, PC DBMS, 4GL)
4. As necessary design source documents.

Contents

- Interface Design Tasks
- User Interface Design
- Input Design
- Output Design

Taxonomy for Computer-Generated Outputs

- Outputs can be classified according to two characteristics
  - Distribution and Audience for Outputs
    - Internal Output (reporting)
    - Turnaround output (external; then internal)
    - External Output (transactions)
  - Implementation Methods for Outputs
    - Printed output
    - Screen output
    - Point-of-sale terminals
    - Multimedia
    - E-mail
    - Hyperlinks
    - Microfilm or microfiche

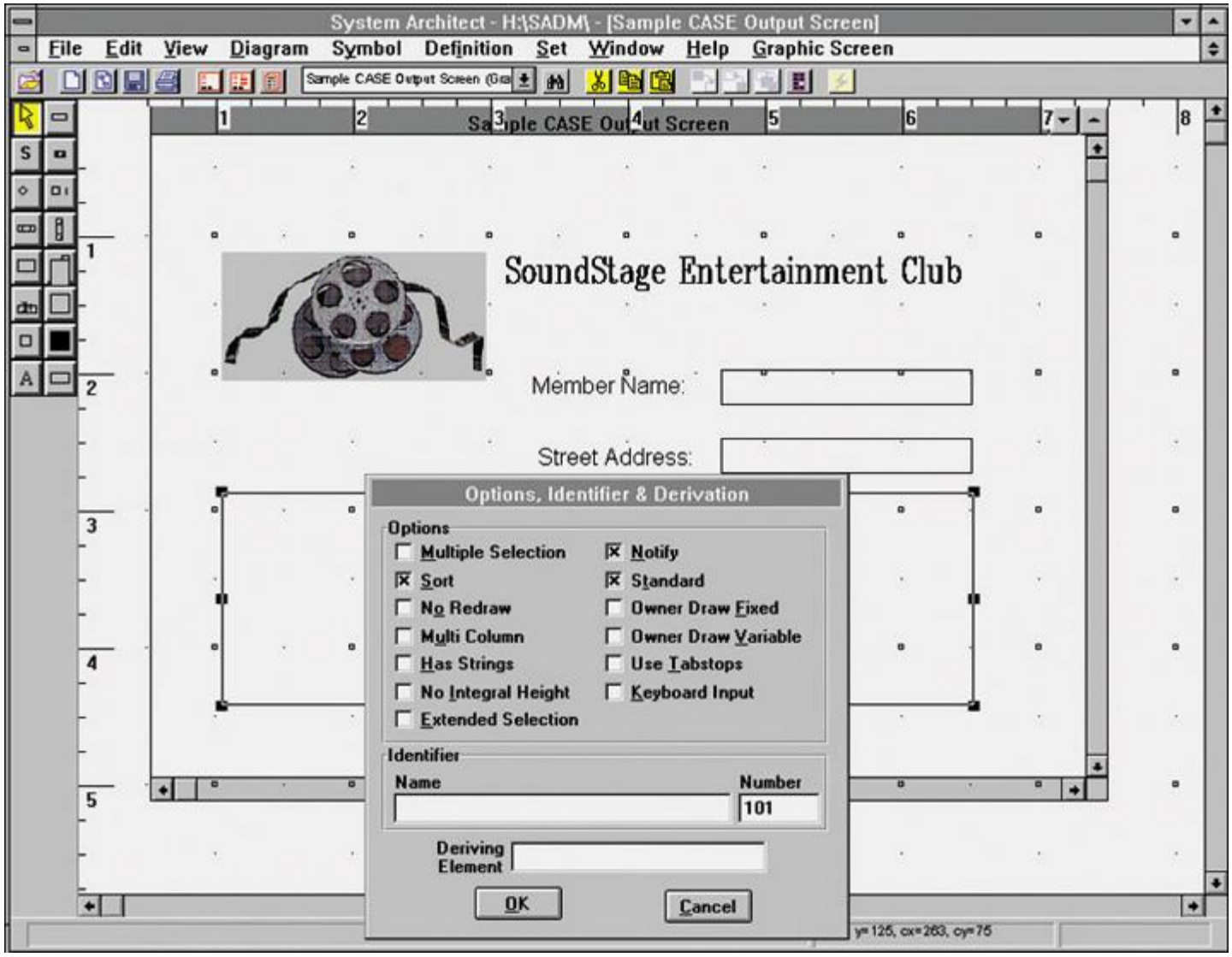
Internal Outputs

- Internal output – an output intended for system owners and system users within an organization.
  - Detailed report – an internal output that presents information with little or no filtering
    - Example: A listing of all customers
  - Summary report – an internal output that categorizes information for managers
    - Do not have to wade through details.
    - Increasingly presented in graphical formats using charts
    - Example: A count of customers by region
  - Exception report – An internal output that filters data to report exceptions to some condition or standard.
    - Example: A listing of customers with past due accounts

External Outputs

- External outputs – an output that leaves the organization organization.
  - Intended for customers, suppliers, partners, or regulatory agencies.
- Turnaround documents – an external output that may re-enter the system as an input.
  - Most “bills” and invoices include a stub to be returned by the customer with payment.

Output Design with a Modern CASE Tool



Output Design Guidelines

1. Outputs should be simple to read and interpret.
   - Include a title.
   - Date and time stamp.
   - Include sections and headings to segment information.
   - Clearly label all fields and columns.
   - Include legends for all abbreviations.
   - Include only required information. Online provide methods to expand and contract information.
   - Report information in format that does not have to be manually edited.
   - Information should be balanced across the page or screen.
   - Provide for easy navigation.
   - Avoid computer jargon and error messages.

Output Design Guidelines (cont.)

2. The timing of outputs is important.
   - This can affect how the output is designed an implemented
3. The distribution of (or access to) outputs must be sufficient to assist all relevant users.
   - The choice of implementation method affects distribution
4. Outputs must be acceptable to the system users who will receive them.
   - Systems analyst must understand how the recipient plans to use the output

Output Design Process

1. Identify system outputs and review logical requirements.
2. Specify physical output requirements.
3. As necessary, design any preprinted forms.
4. Design, validate and test outputs using some combination of:
   a) Layout tools (e.g., hand sketches, spacing charts, or CASE tools.
   b) Prototyping tools (e.g., spreadsheet, PC DBMS, 4GL)
   c) Code generating tools (e.g., report writer)