

Enterprise Application Integration (EAI)

Chapter 5 User Interface-Level EAI

Introduction

- User interface-level EAI, the most primitive of all EAI levels, is also the most necessary. Other EAI levels might be more technologically appealing and more efficient, but the simple reality is that, in many applications, the user is the only available mechanism for accessing logic and data. In spite of its inefficiency in "getting into" an application, the user interface has an advantage in that it does not require any changes to the source or target application

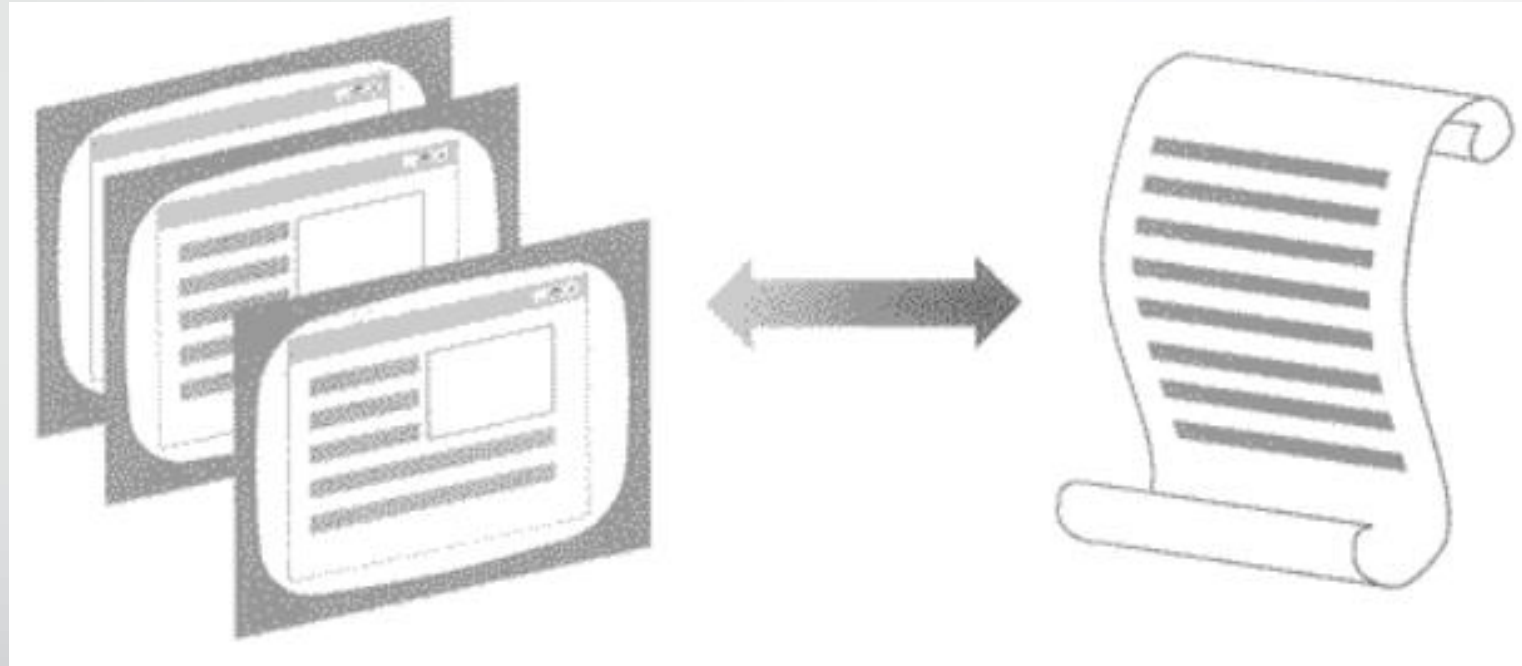
Going to the User Interface

- If performed correctly, there will be virtually no difference between using the user interface or a true application interface
- It is necessary to know how the application data and methods fit within the enterprise metadata model and the enterprise object models, and map the source system into this logical and physical layer. Using the user interface as a point of integration does not free us from this requirement.

Understanding the Application

- Understanding the **underlying data storage schema**, much of the **application logic**, and, most importantly, **how the information is presented to the user interface**
- Unlike other interface levels, information presented to a user interface may not map back to a database.
 - Most of the data elements on the screen, such as calculated fields (e.g., an order total), are created by the application logic and do not come directly from the database.
 1. Requires reading the documentation
 2. The applications are not always well documented
 3. The source code itself will have to be read

Creating the Screen Catalog



- Mapping Screens

Finding the Information

- There are two basic techniques to extract information from screens:
- **Static screen extraction**

The information is being accessed from a static point on the screen (e.g., Column 10, Row 12, Position 10).

- ✓ Simplicity
- ✓ Going to the predetermined location and grabbing the information
- If the application interface changes for any reason (e.g., by adding more information) then ...

Finding the Information (cont.)

- **Dynamic extraction** is a more sophisticated process than static extraction.
Rather than going directly to a fixed position when looking for accounts receivable information, this technique would search for the "accounts receivable " label and would use that as a reference point
 - ✓ Automatically adapt to changes
 - ✓ The ability to place specific logic sequences in the extraction process, such as a logic to flag errors if a particular label is not found, or one that determines if a data element does not make sense

Finding the Information (cont.)

- **Error Processing**

- Both the dynamic and the static extraction techniques must go through a rudimentary error-checking routine in order to ensure that only valid data is extracted.
- Error-processing routines are vital to user interface-level EAI

Approaches

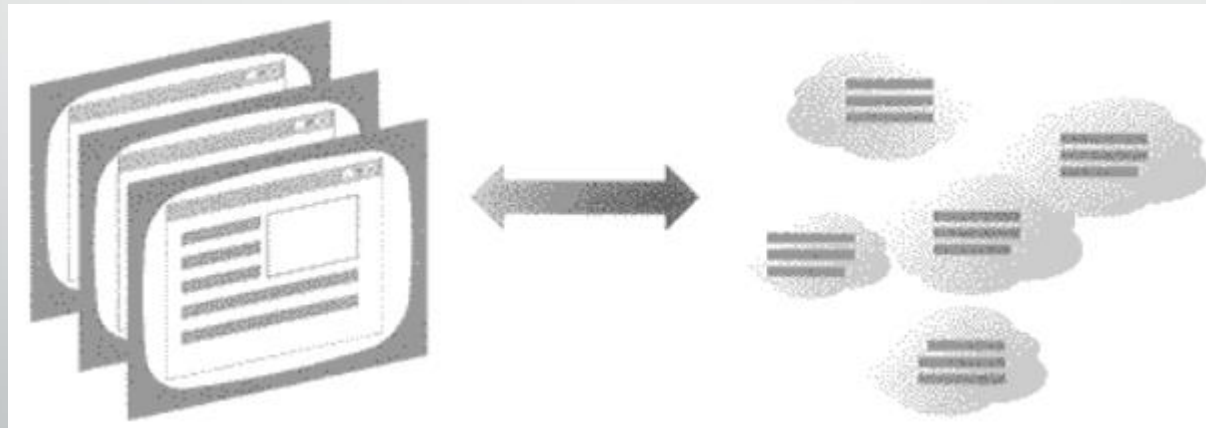
- There are two approaches for getting information from application interfaces:
 - Using screens as raw data
 - Using screens as objects
- Both extract the information by using the interface as if the program was an actual user
- It is impossible to get all necessary information from a single user interface. Most user interface-level EAI efforts will require as many as a thousand instances of a user interface in order to obtain the correct information in the correct volume

Screens-as-Data

- Looking at a screen as a simple stream of text
- The information is parsed, identified, converted, and processed within the program that is responsible for processing the user interface information

Screens-as-Objects

- Too sophisticated for most user interface-level EAI projects
- Using screens-as-objects requires translating the information gathered from a user interface into an application object



Screen Access Tricks

- There are three technical approaches to extracting information from user interfaces:
 - HLLAPI (High Level Language Application Program Interface)
 - ASCII or ANSI terminals
 - OLE Automation