

Event-Based System Design

Session 560

Mike Bowers
Texas Instruments



Agenda

- Overview of the process
- Benefits
- Detailed description defining systems based on events
- Conclusion

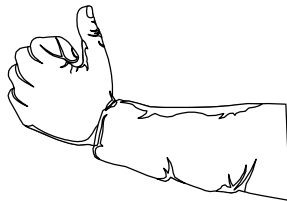


Event-Based System Design

- Issues facing all development teams
 - How to build a system that fits the needs of the user
 - How to build the system rapidly
 - How to involve the user
- A solution!
 - Event-based system design



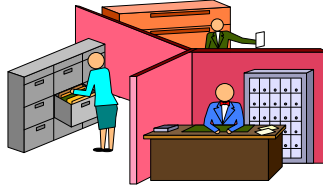
Benefits—Happier Users



- Business users can do this activity with little training
- More involvement by users
- Greater confidence by the users
- More enthusiasm by users



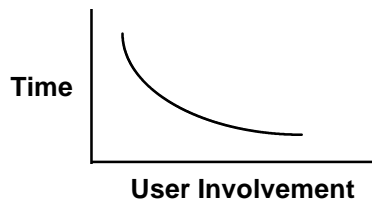
Benefits–Tailored Applications



- Each group of users gets an application that meets their needs
- Tailored applications are more effective solutions
- Tailored applications are easier to develop – less complicated



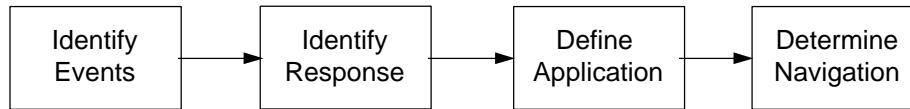
Benefits–Faster Development



- Emphasis is on rapid development
- Events are easily identified
- Response paths are easily identified
- Results in faster system design



Event-Based System Design

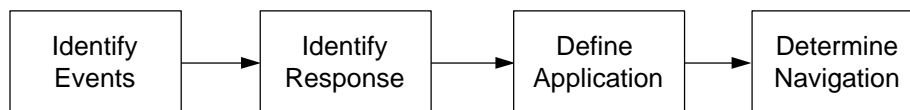


Process:

- Step 1 - Identifying the Events
- Step 2 - Identifying the Response Path
- Step 3 - Defining the Application
- Step 4 - Determining the Window Navigation



Step 1. Identifying the Events



- Identify who initiates events in the business environment and who responds to them
 - External actors
 - Internal actors
- Identify the events for each external actor
- Systems are built for actors



Actors

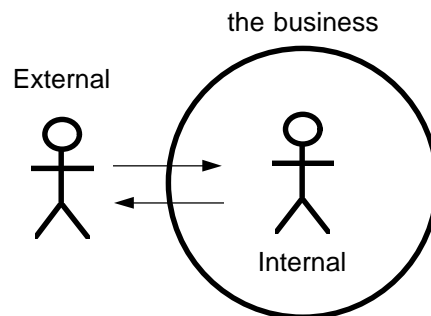


- People who interact within the context of the business
- Are the initiators of activities (events) within a business
- Respond to events

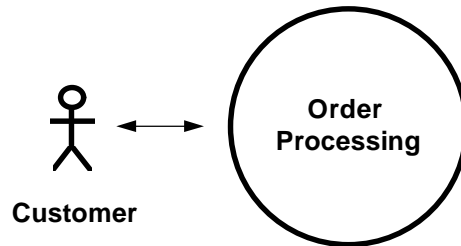


Types of Actors

- Types of actors
 - External actors
 - » Trigger the event
 - Internal actors
 - » Respond to the event



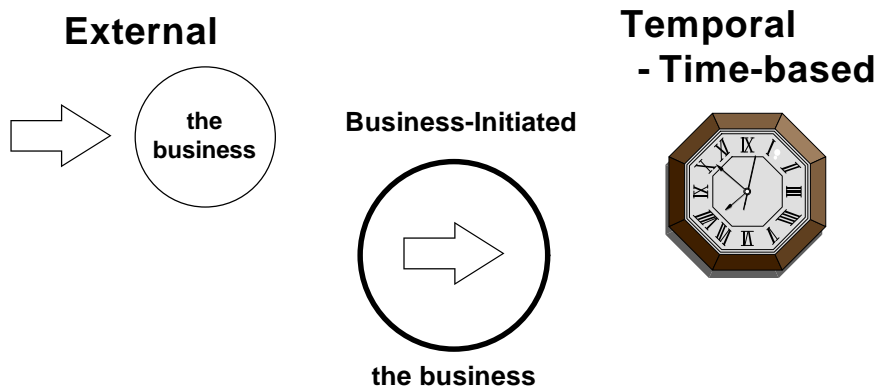
1a–Identifying the External Actors



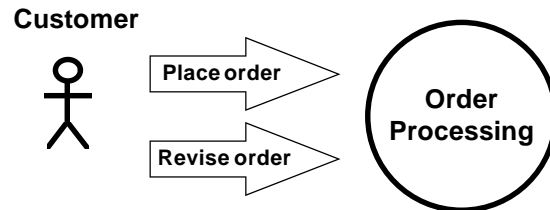
- Identify the external actors to the business area
- Define who will be initiating events that the system will respond to



Types of Events



1b–Listing Events by Actor

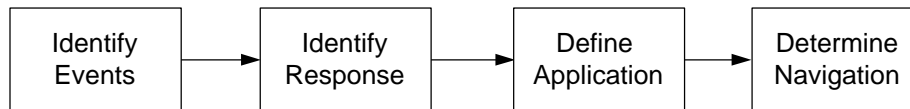


For each actor:

- List each event within the business area. Don't define the response path at this time, the emphasis is on speed.
- Define the beginning of the event
- Define the desired response to the event



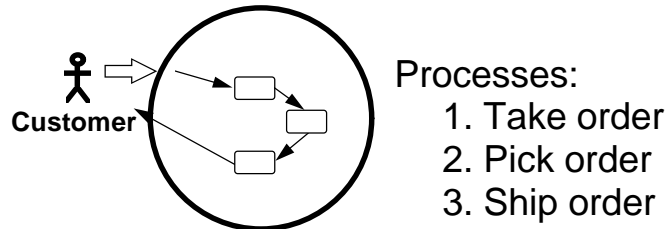
Step 2. Identifying Event Response Paths



- Determine the response path
- Decompose the event if necessary
- Why do this? The response path will define the system design.



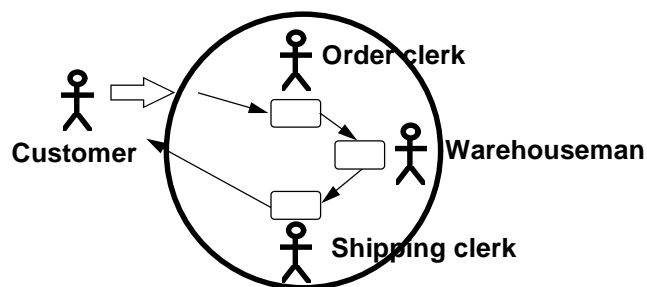
2a–Determining the Response Path



- For each external actor:
 - For each event:
 - » Determine the steps necessary to get from the beginning to the end
 - » Include displays of information or lists as well as create, update, and delete activities



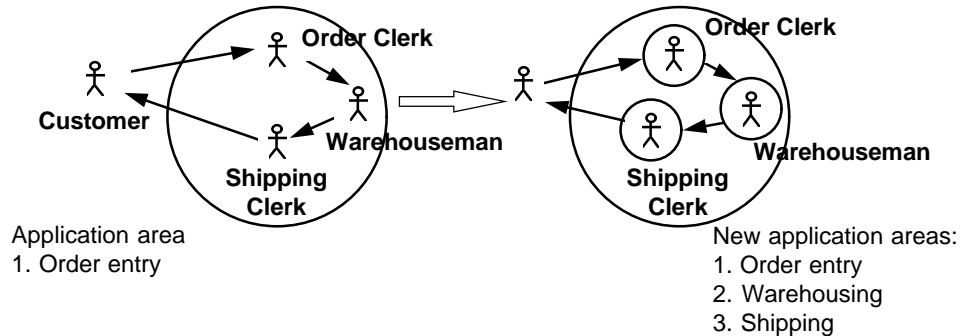
2b–Determining the Actors Involved



- Identify the actors involved in each step
- If more than one actor is involved in the response, decompose the event



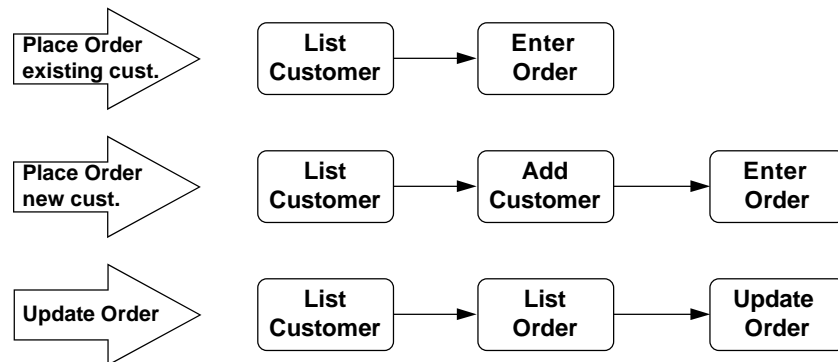
Decomposing Events



- Create new application areas by actor
- Return to step 1
- Benefit: separate application areas allow tailored systems



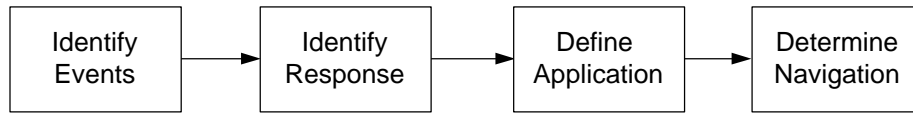
Events for Order Entry



- List each event and the response path
- Include lists and displays as necessary



Step 3. Defining the Application

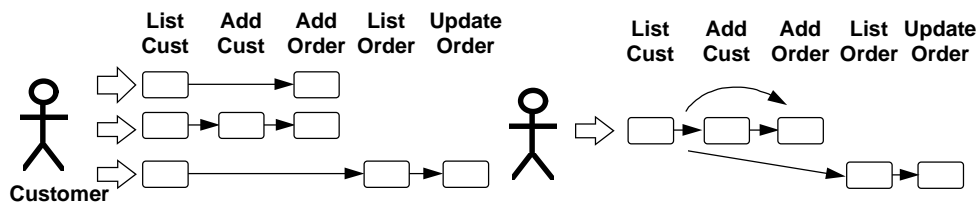


- Determine common actions in an application area
- Define the application



3a—Consolidating the Response Paths

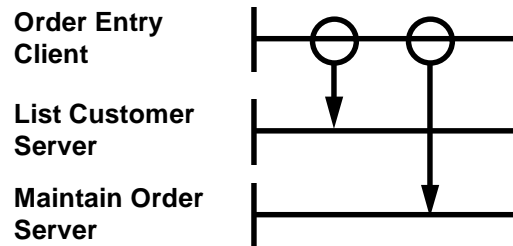
Application Area - Order Entry



- Begin from each external actor and consolidate the activities initiated by that actor
- Take into consideration security issues and define separate windows if necessary



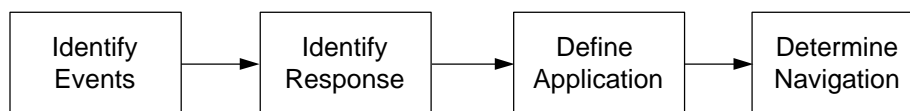
3b–Defining the Application



- Define the business system to Composer
- Define the procedures necessary to support the system



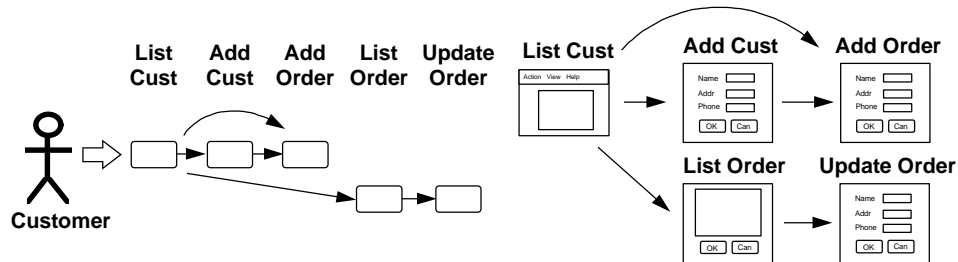
Step 4. Determining the Window Navigation



- Define the window navigation
- Define scenarios for testing



4a–Defining the Window Navigation



- Navigation driven by the response path diagrams
- Each step of the response may result in window or dialog box
- Additional dialog boxes may be required for lookups



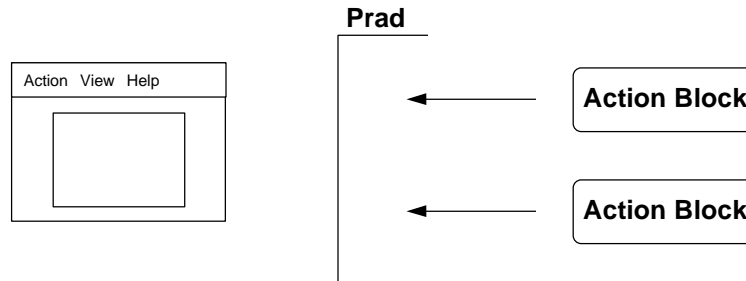
4b–Developing Scenarios for Testing

- Based on the events, develop use case examples to test the system
- Includes system-oriented actions (push buttons, etc.)
- Test the window navigation
- Revise as necessary
- Benefit: allows the system design to be tested earlier

Scenario 1	
Action	Response
Step 1	
Step 2	
Step 3	
Step 4	
Step 5	



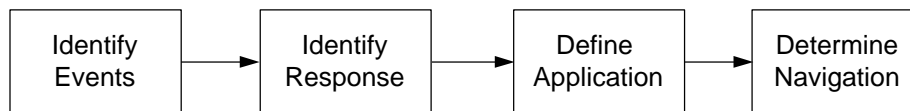
Completing the System



- Complete the interface logic
- Add business rule logic
- Add data management logic



Review of the Process



- Step 1 - Identifying the events
- Step 2 - Identifying the response path
- Step 3 - Defining the application
- Step 4 - Determining the window navigation



Conclusion

Why use event-based system design?

- Events are natural occurrences within a business
- Business people relate to events
- The event response becomes the design of the system
- The results are:
 - Happier users
 - Tailored applications
 - Faster development



Event-Based System Design

Session 560

Mike Bowers
Texas Instruments

