

★ **Research Demonstration
by OpenProcess, Inc.**

A Software Engineering Approach and Tool Set for Developing Internet Applications

Presented by David A. Marca

to the 22nd International Conference on Software Engineering at Limrick, Ireland

Internet-ready business process for maximum control,
on-line learning, real-time monitoring and selective opening

☆ Software Engineering for e-Business

Electronic connections can be defined by assessing their effect on the supply chain.

□ Market Focused

- Link a base of customers to a vertical market of suppliers
- Decreasing marketing costs + increasing orders = lowering prices
- Requires brand recognition of web site and products

□ Production Focused

- Outsource a portion of the value of a business
- Others perform the activity that create the value
- Requires monitoring partner performance

☆ Software Engineering for e-Business

Business factors and rules of engagement for a B2B connection can be managed by both parties.

- ❑ Use a software engineering approach and tool set.
- ❑ Create an architecture of the business relationship.
- ❑ Develop adaptable processes within that architecture.
- ❑ Support both with an adaptable technical infrastructure.

All designs constantly mirror and support the business partnership.

☆ Software Engineering for e-Business

Adaptable e-Business connections arise when all of the following are created in order:

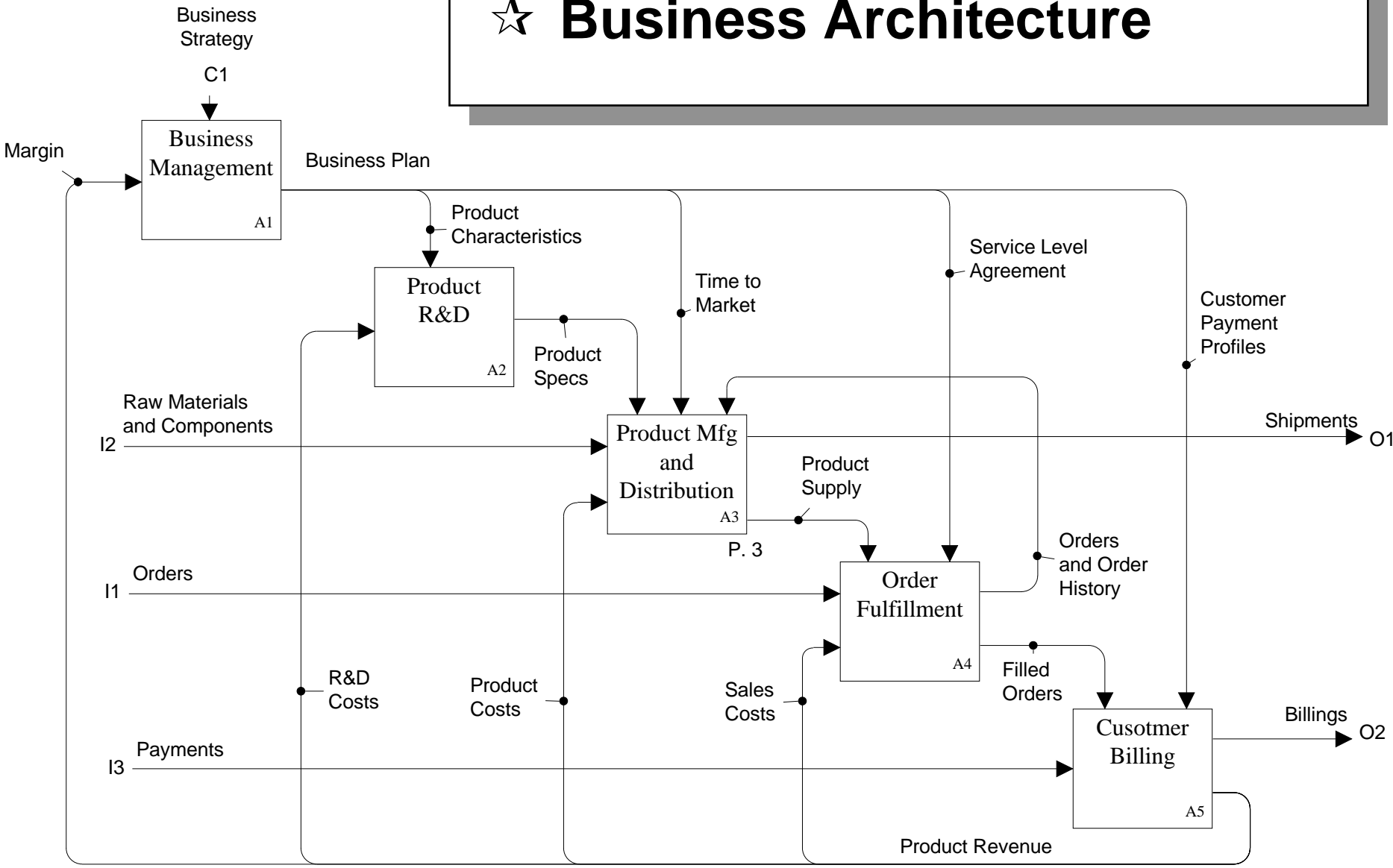
1. **Business Architecture** for the partnership(s).
2. **Business Design** that defines all essential policies.
3. **Process Architecture** designed for adaptability.
4. **Process Design** for the activities of the connection.
5. **Technical Architecture** that enables adaptability.
6. **Technical Design** that supports process activities.

#1 Business Architecture Business Structure + Rules

Define boundaries around major business units, fundamental policies and external market forces.

- **Boundary defines the roles for each business unit.**
- **Policies establish cycle times.**
- **External Forces are events that affect roles and cycle times.**
- **Software Engineering principles used:**
 - Modularity
 - Well-defined Subsystem Interfaces

☆ Business Architecture

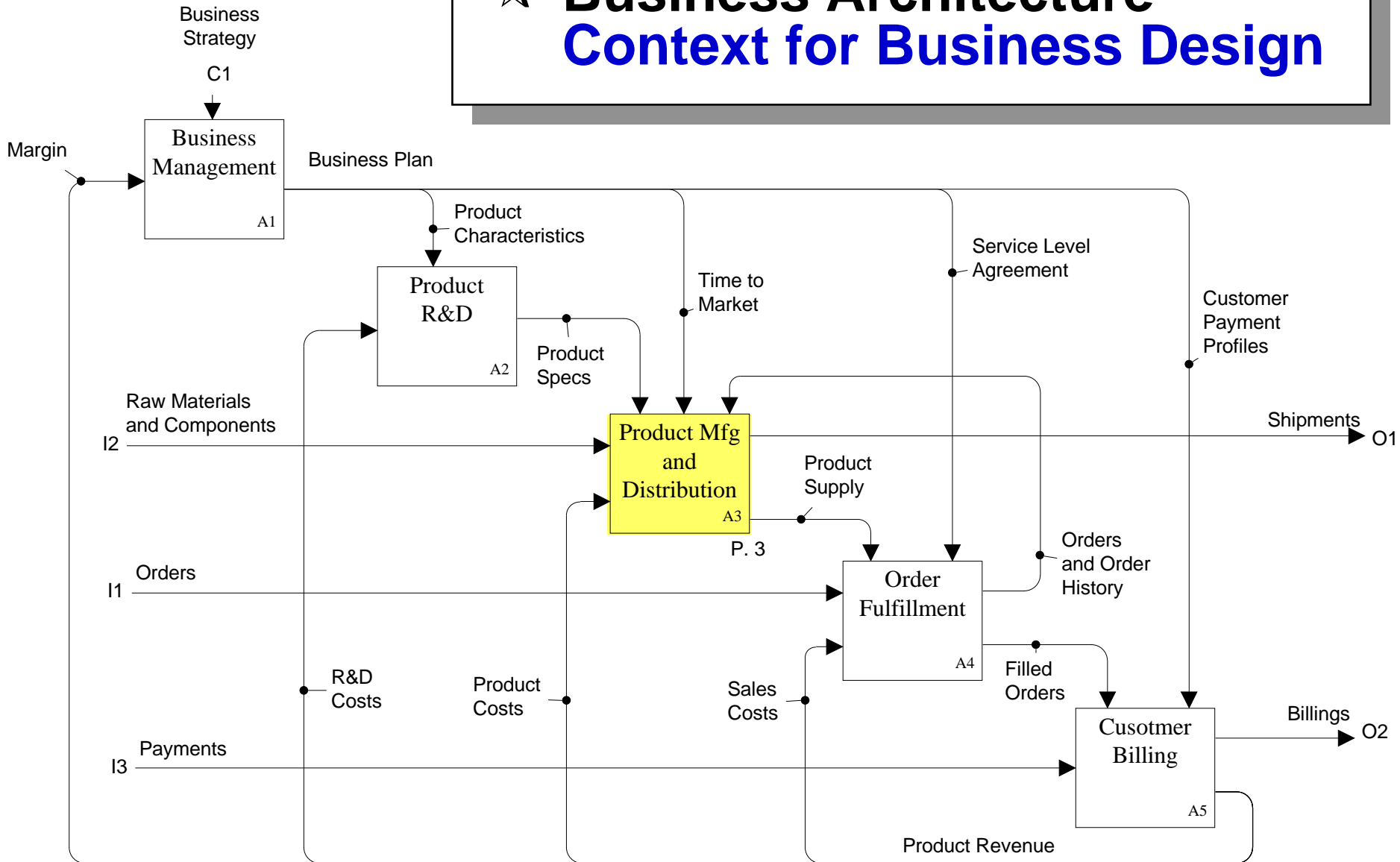


#2 Business Design Events/Responses + Policies

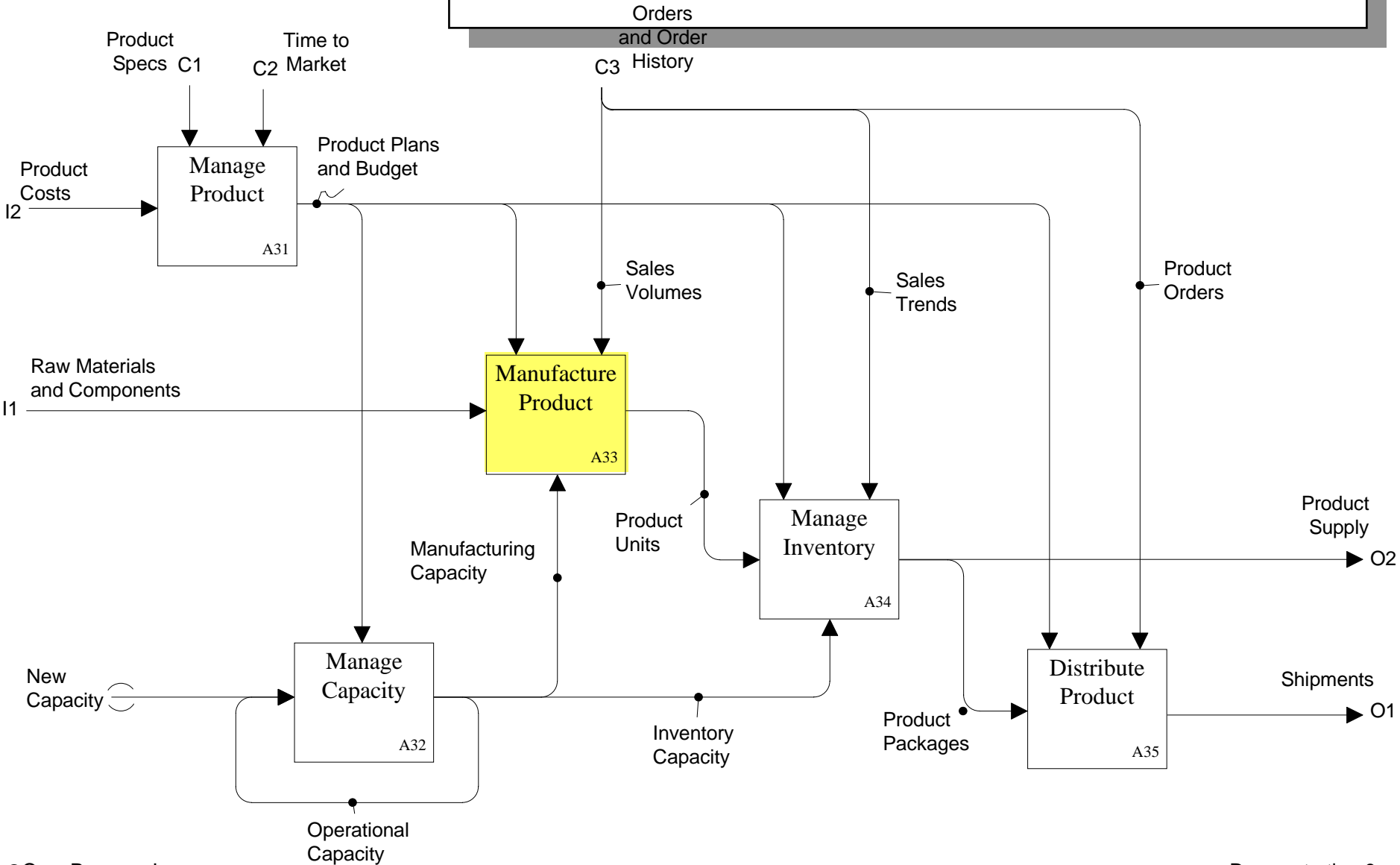
Define how each business unit responds to each external business event it receives.

- **Context is one box on the Business Architecture model.**
- **Events = input arrows. Responses = output arrows.**
- **Policies that govern how to respond = control arrows.**
- **Software Engineering principles used:**
 - Event-Response Design
 - Requirements Definition -- for Service Level Agreements (SLAs)

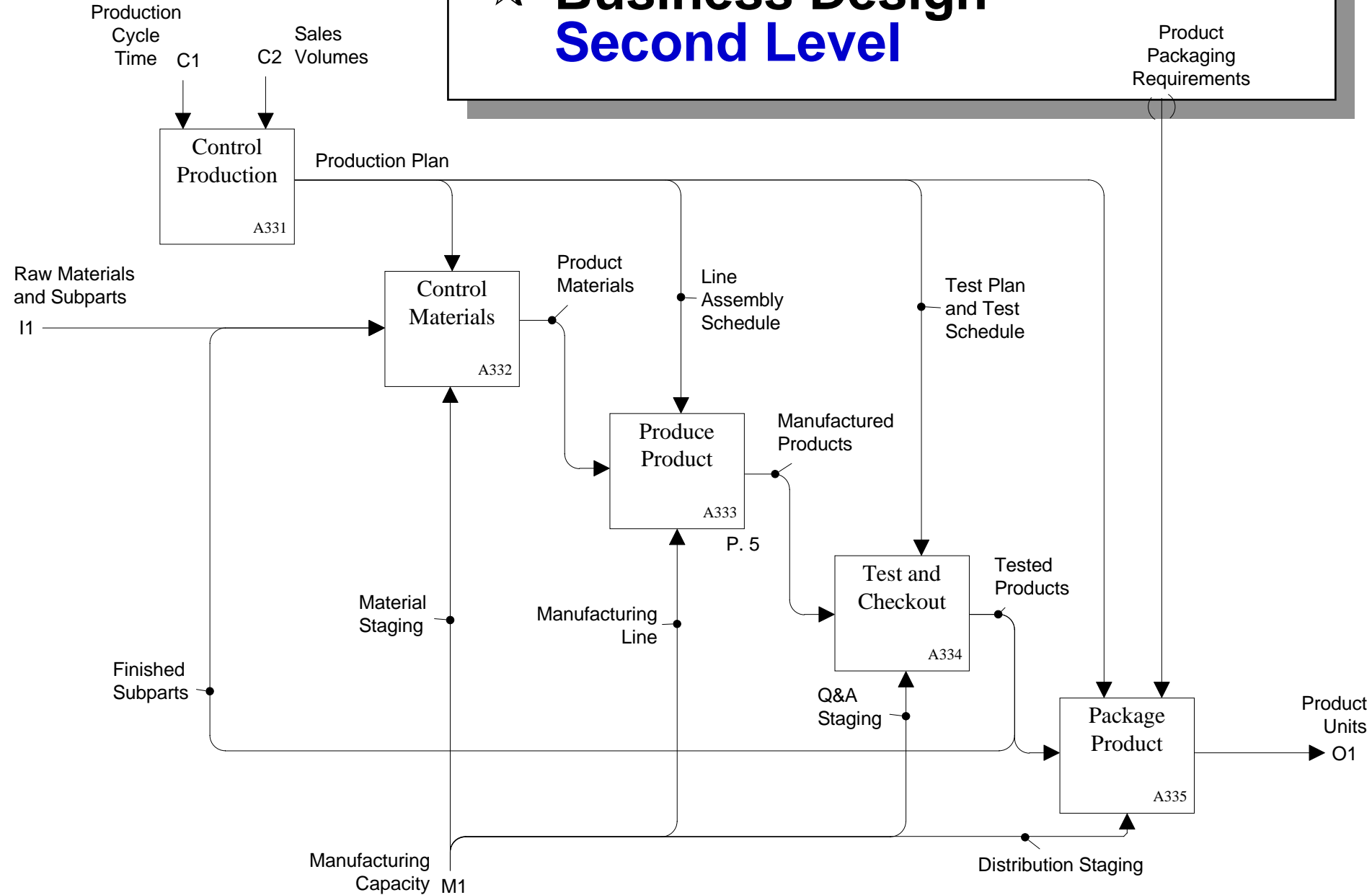
☆ Business Architecture Context for Business Design



☆ Business Design First Level



★ Business Design Second Level

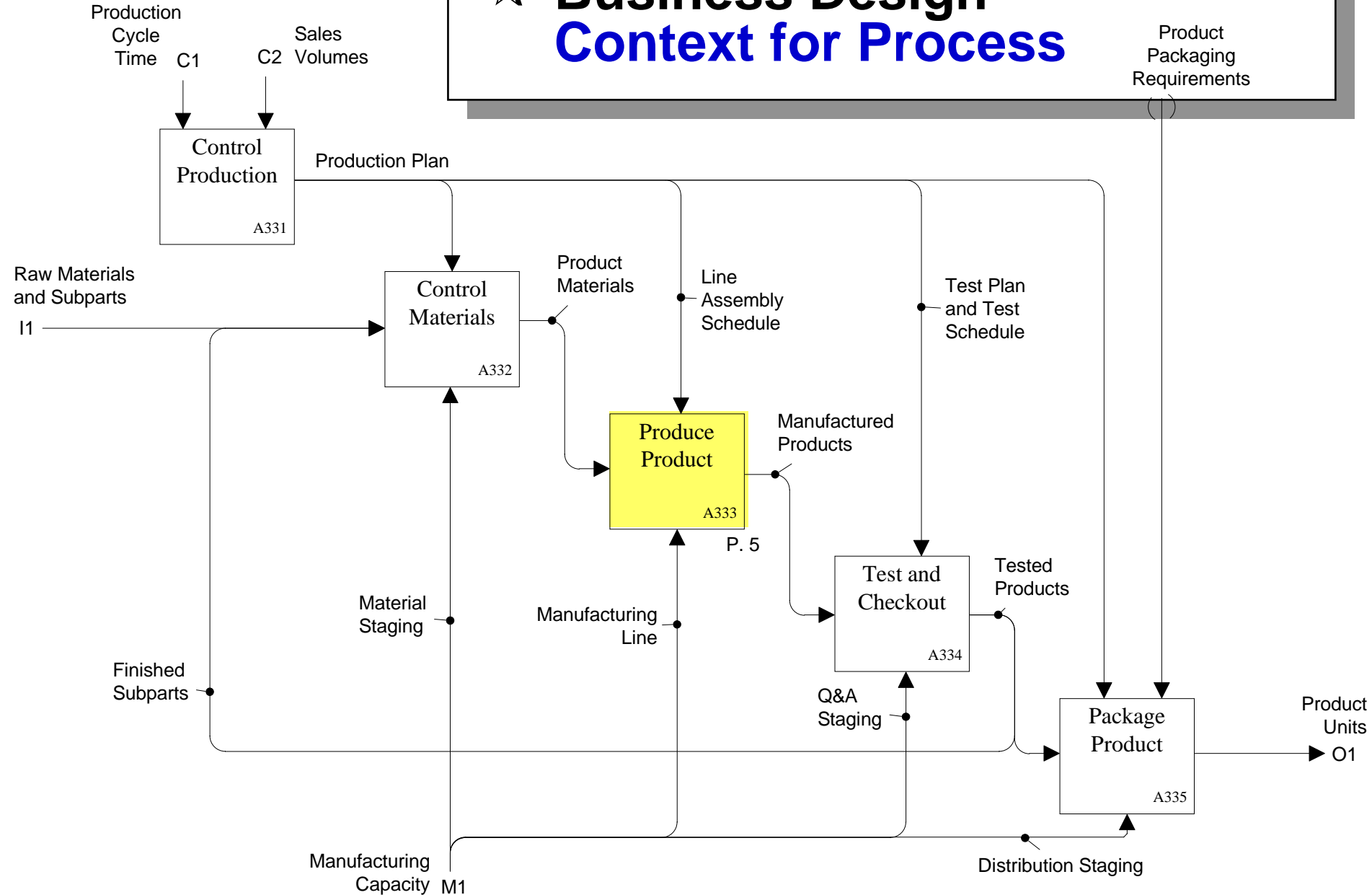


#3 Process Architecture **Adaptable Operations**

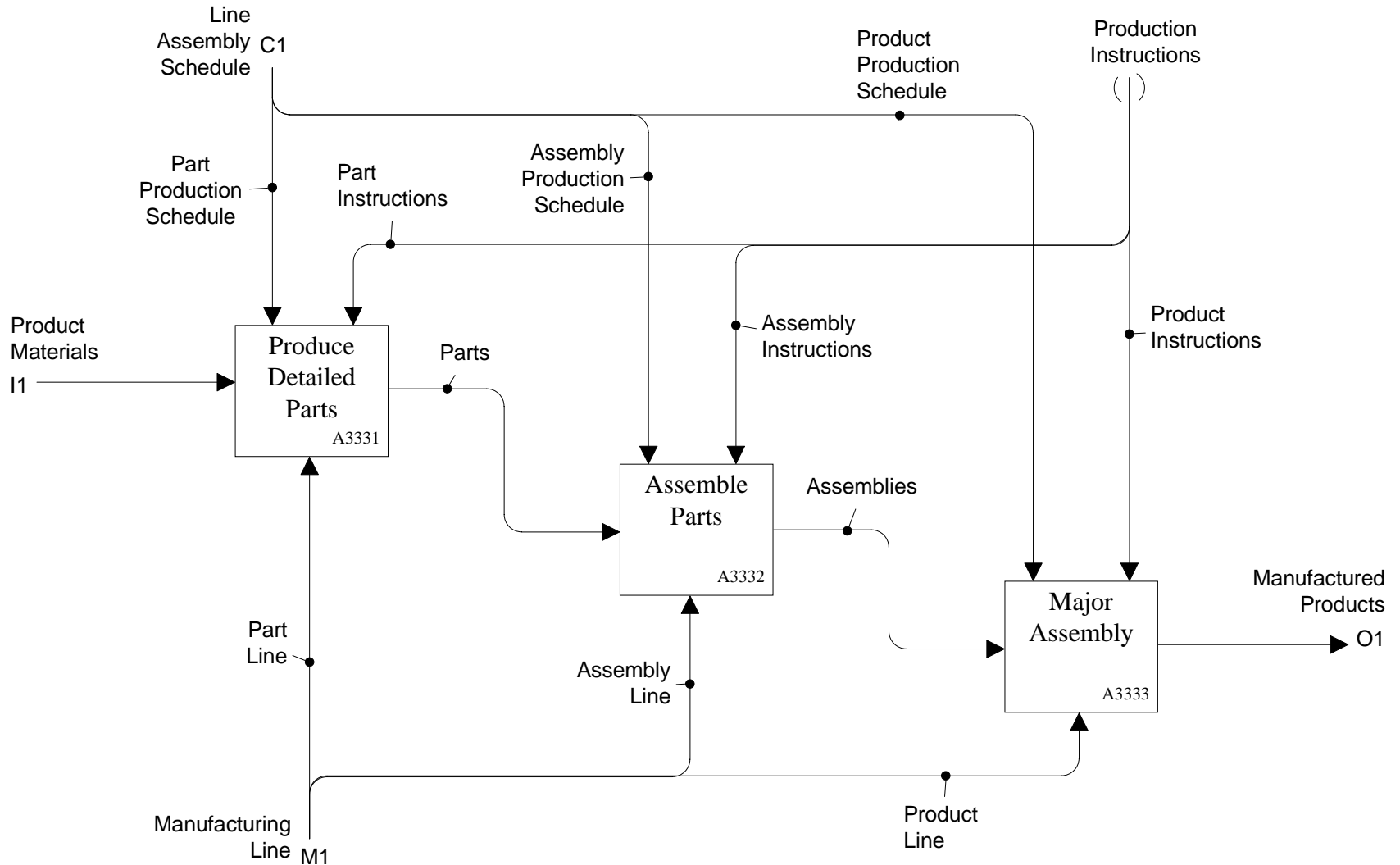
Ensure each party can adapt its operations to changing: events, cycles, market conditions.

- ❑ **Context is one box on the Business Design model.**
- ❑ **Decompose to identify the atomic events and responses.**
- ❑ **Operating Steps create responses to events within service level agreements.**
- ❑ **Software Engineering principles used:**
 - Data Transforms
 - Coupling / Cohesion

★ Business Design Context for Process



☆ Process Architecture

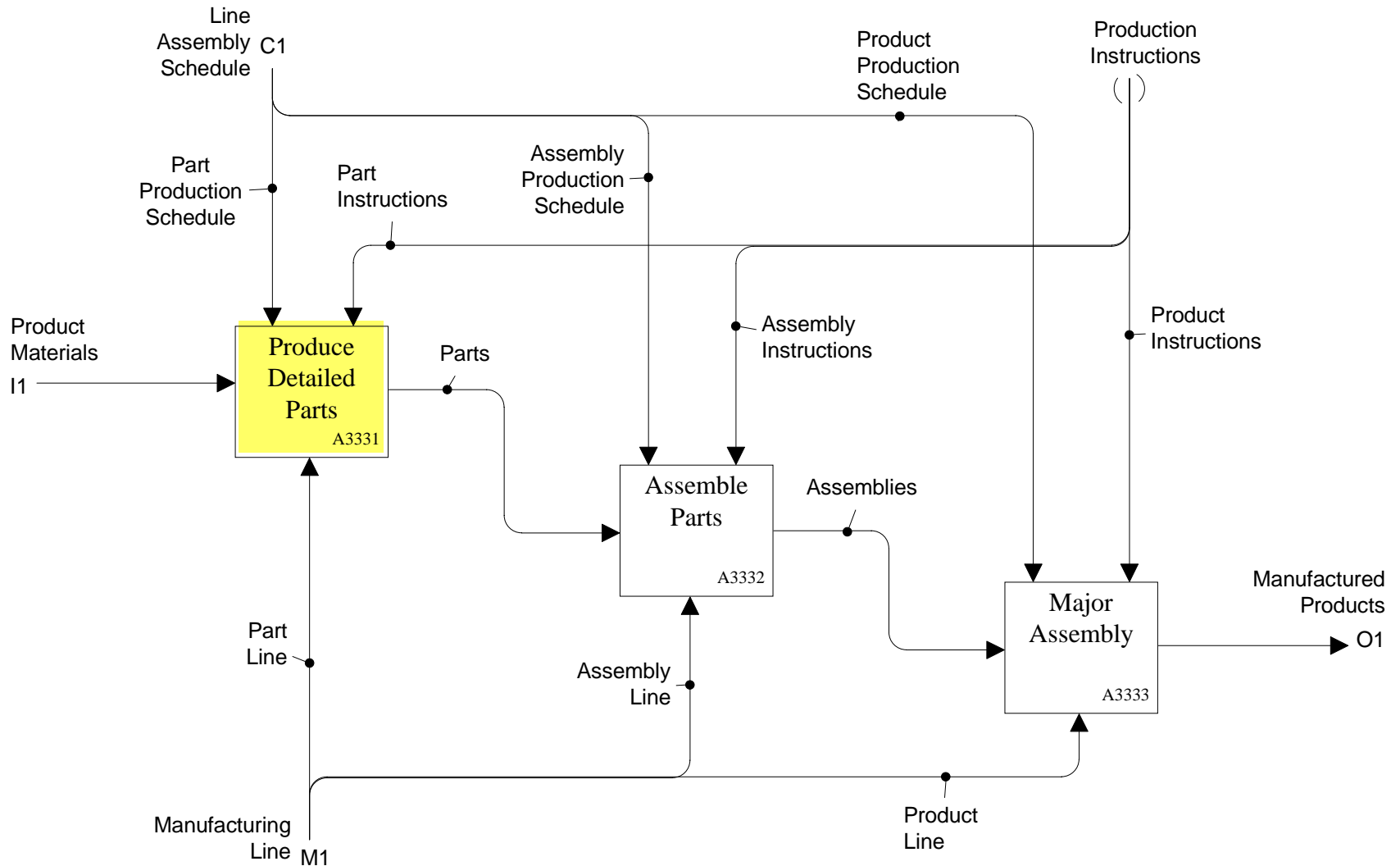


#4 Process Design Completeness + Consistency

Define accountabilities, intended outcomes, expected rules of engagement, response times.

- ❑ **Process Description** for each operational step.
- ❑ **Process Template** categorizes and provides consistency.
- ❑ **Inter-linked Hypercode** versions of all process descriptions.
- ❑ **Software Engineering principles used:**
 - Pseudo-code
 - Data Repository Design

☆ Process Architecture Context for Process Design



☆ Process Design

The screenshot displays the 'OpenProcess(tm) Framework Builder Version 2.0' application. The main window is titled 'A0' and 'Company XYZ'. The interface is divided into several sections:

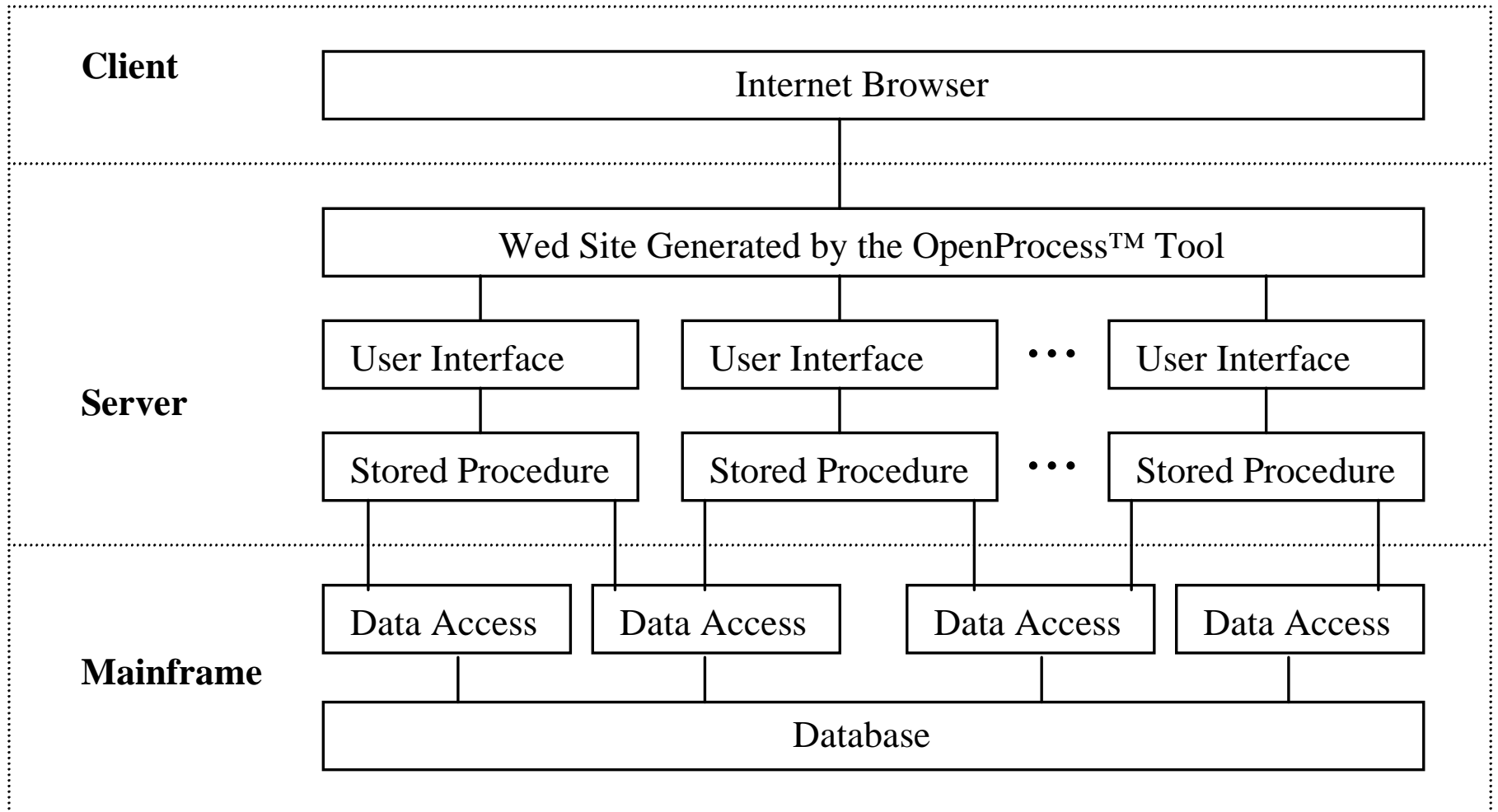
- Policy:** Contains a list of process policies:
 1. Business Strategy determines what Products to make.
 2. Orders determine how much Raw Materials and Subparts to order.
 3. Billings match, one-for-one, Shipments.
 4. Customer Payments are matched to their respective Billings.
- Exceptions, SLA, Roles:** These sections are currently empty.
- ISO9000, Controls, Metrics, Contacts:** These sections are also currently empty.
- Inputs:** A list of process inputs including: Orders, Payments, Raw Materials and Subparts, and Product Revenues.
- Links:** A list of links pointing to glossary files:
 - file:///C:/process/ICSE/Glossary/Orders.txt
 - file:///C:/process/ICSE/Glossary/Payments.txt
 - file:///C:/process/ICSE/Glossary/Raw Materials and Subparts.txt
 - file:///C:/process/ICSE/Glossary/Product Revenues.txt

#5 Technical Architecture Mirrors the Process

Define thin-client solution, preserve boundaries, isolate policies and break up applications.

- ❑ **Dedicated Server for each business unit, for each event-response pair.**
- ❑ **Process and Data Allocation across the computer topology.**
- ❑ **Small GUIs so steps use only the application logic they really need.**
- ❑ **Software Engineering principles used:**
 - Adaptable Client/Server Architecture
 - Graphical User Interface Design

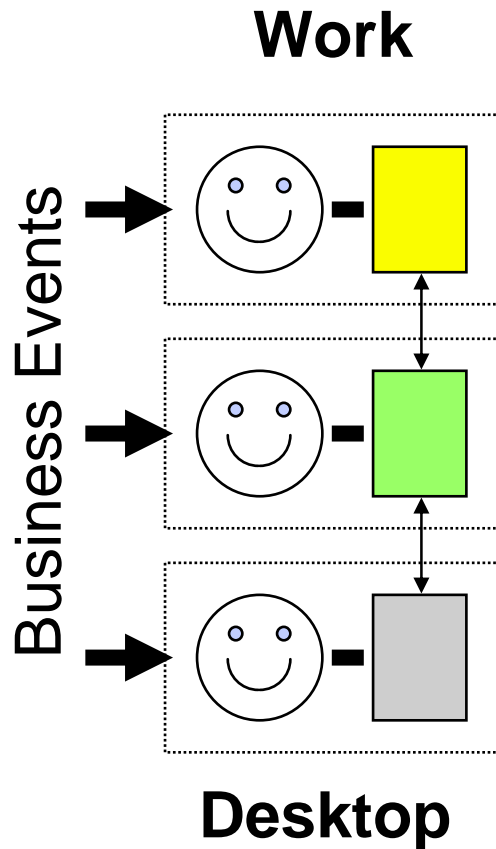
☆ Technical Architecture



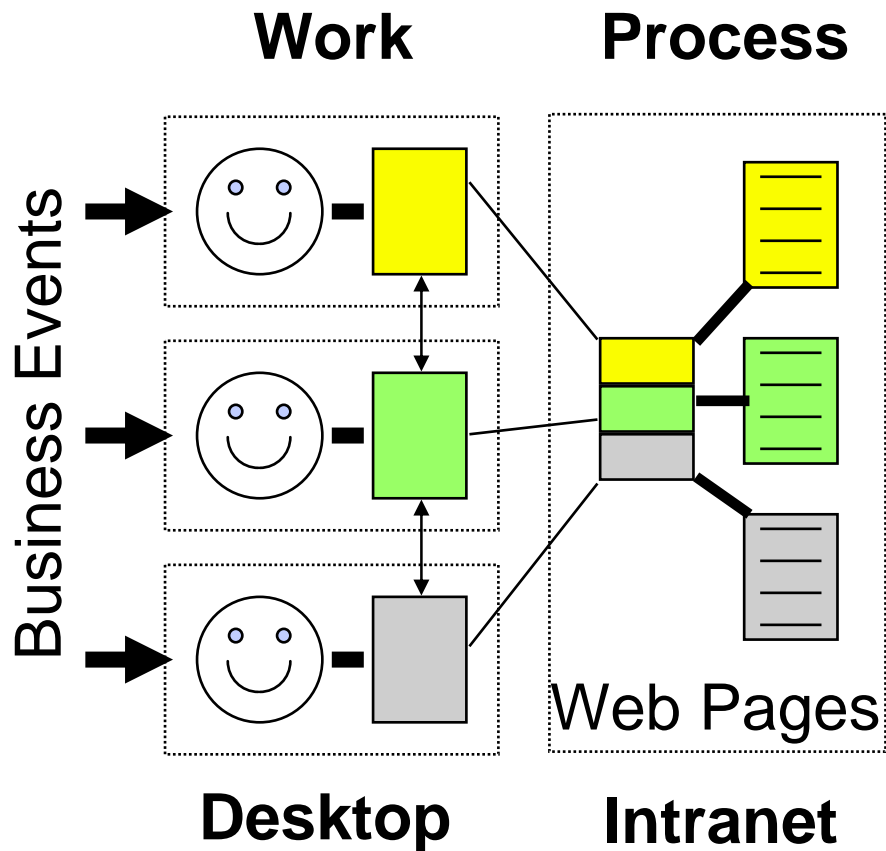
☆ **Each Business Event
Requires a Response**

Business Events →
→
→

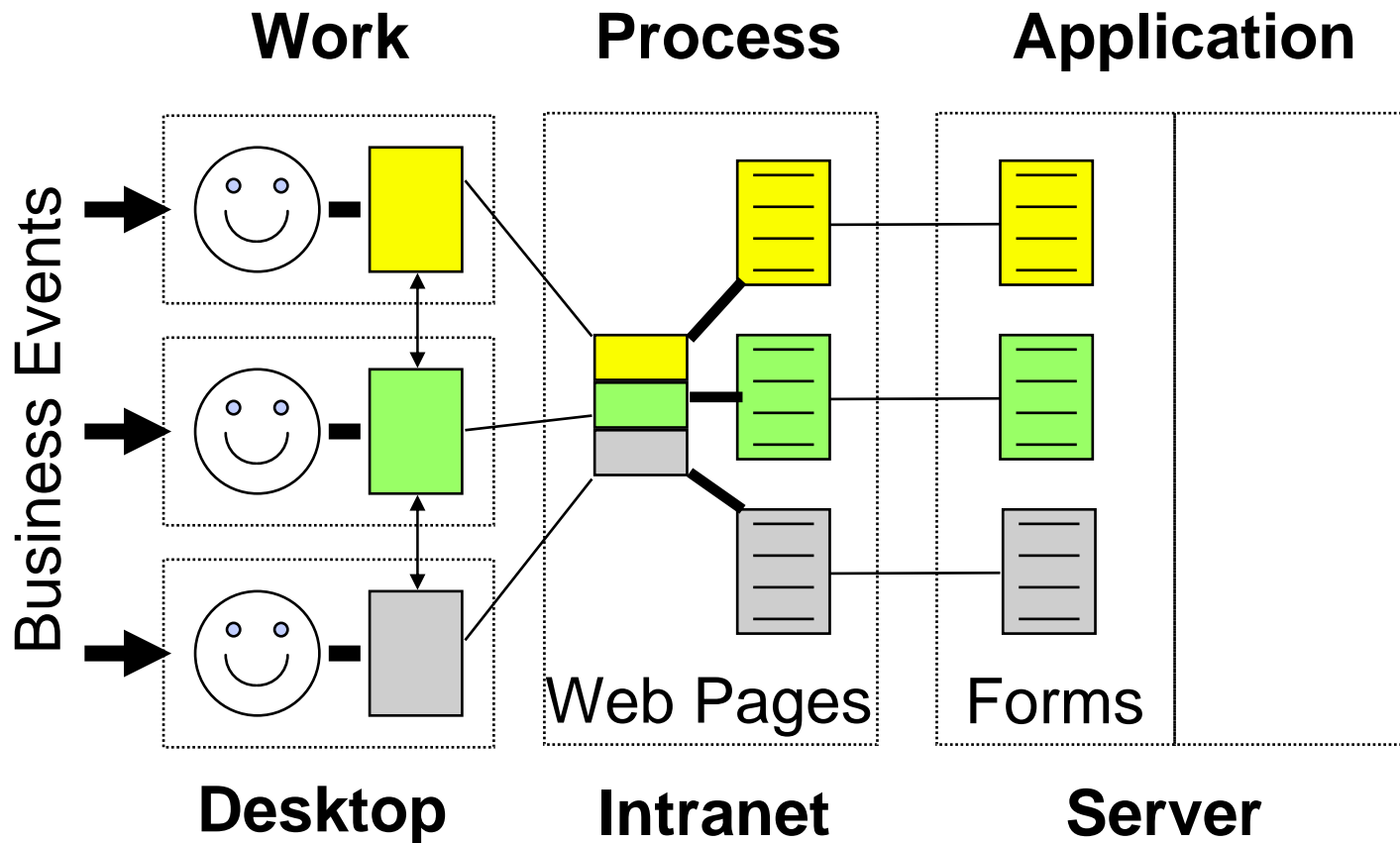
☆ People Respond by Following Their Particular Process



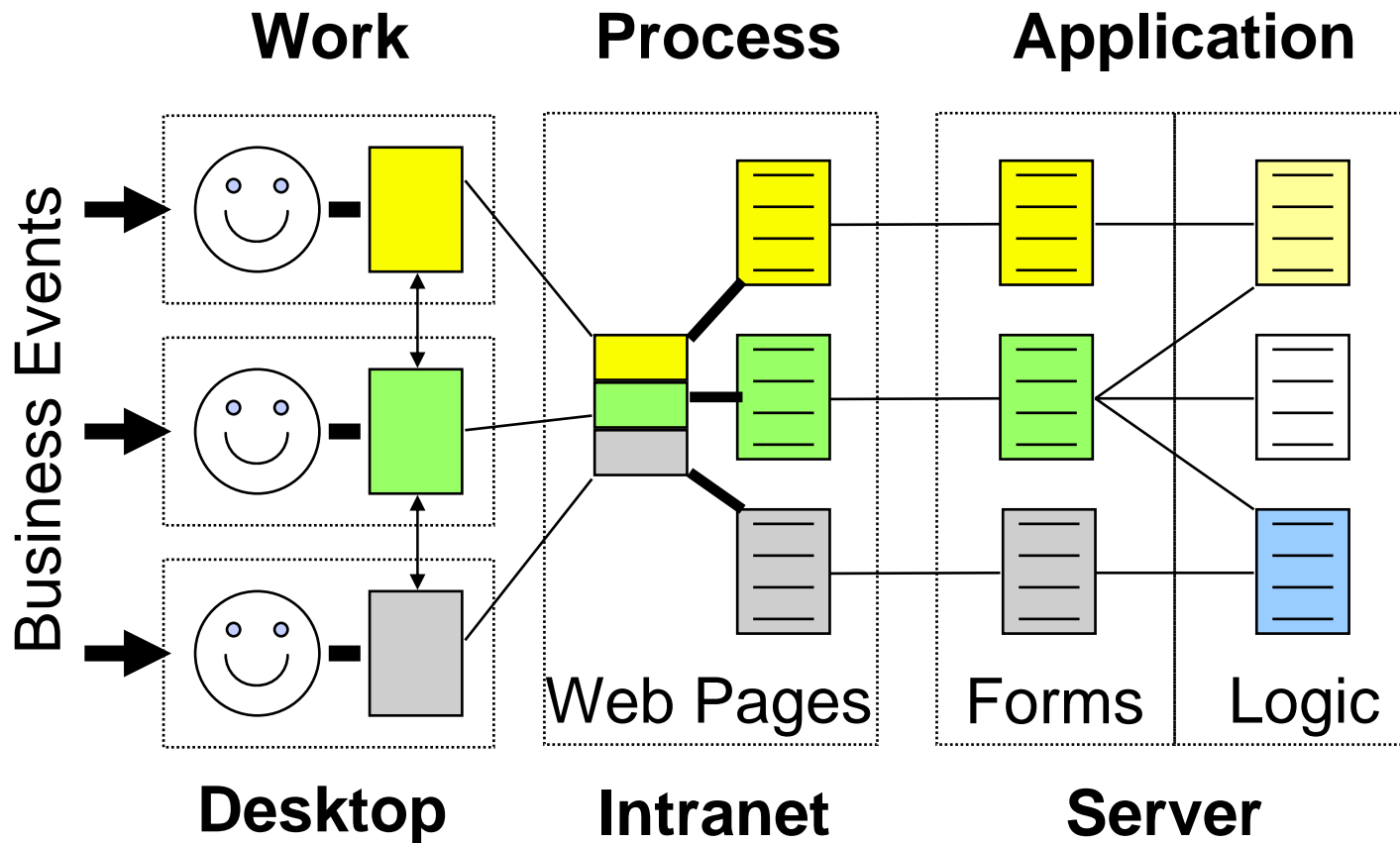
★ That Particular Process is Part of the Whole Process



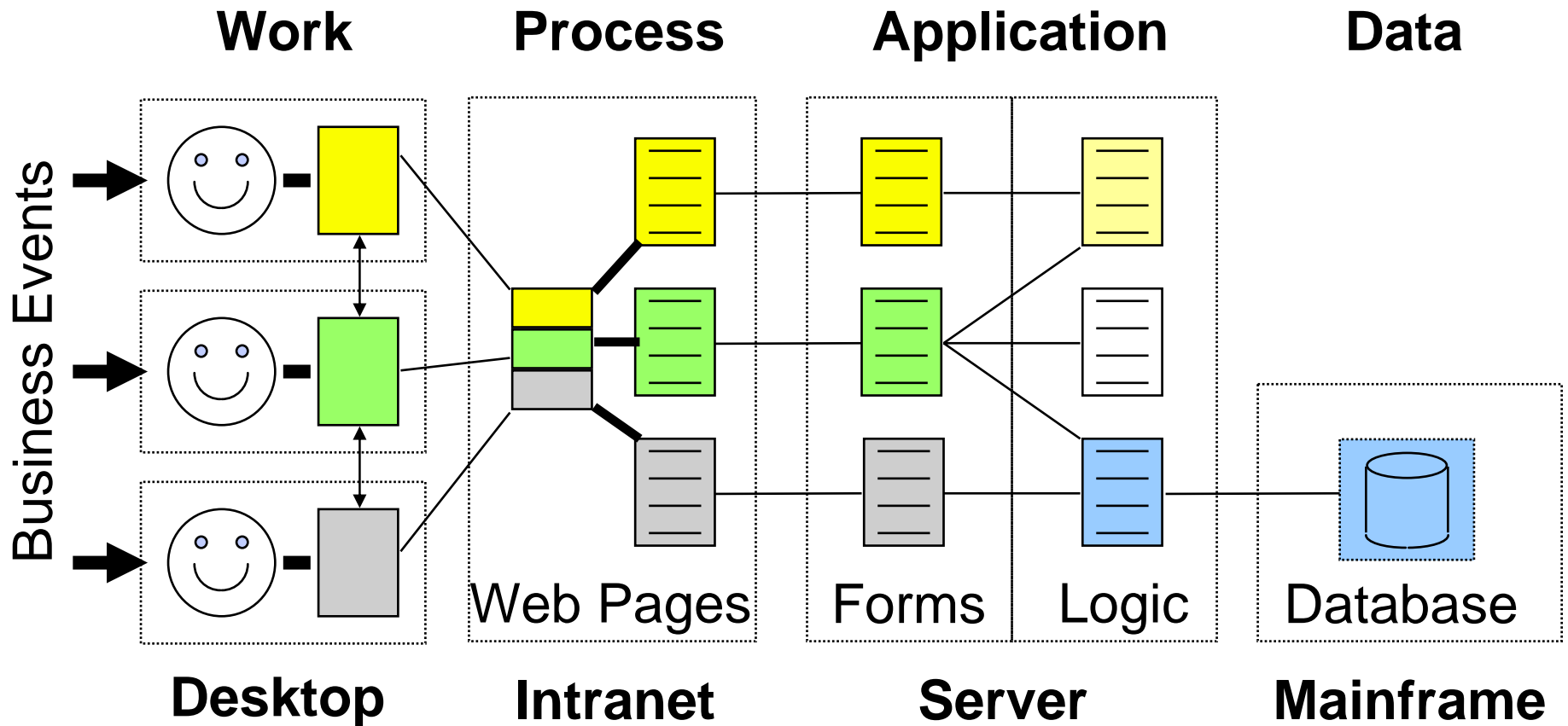
☆ Process is Supported by Applications via Web Forms



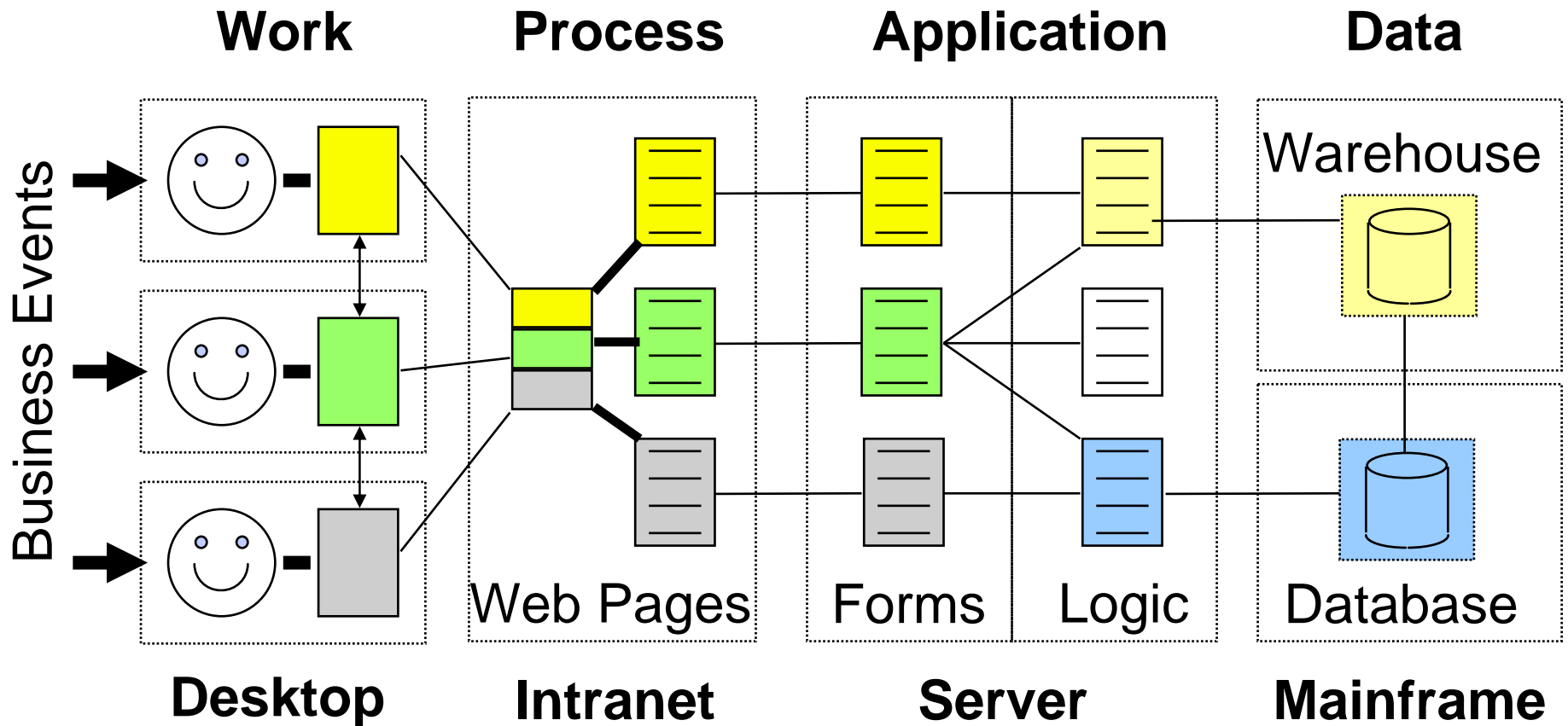
★ Application Logic and Web Forms are Always Separated



★ **Some Application Logic Performs the OLTP**



★ **Some Application Logic Performs the OLAP**

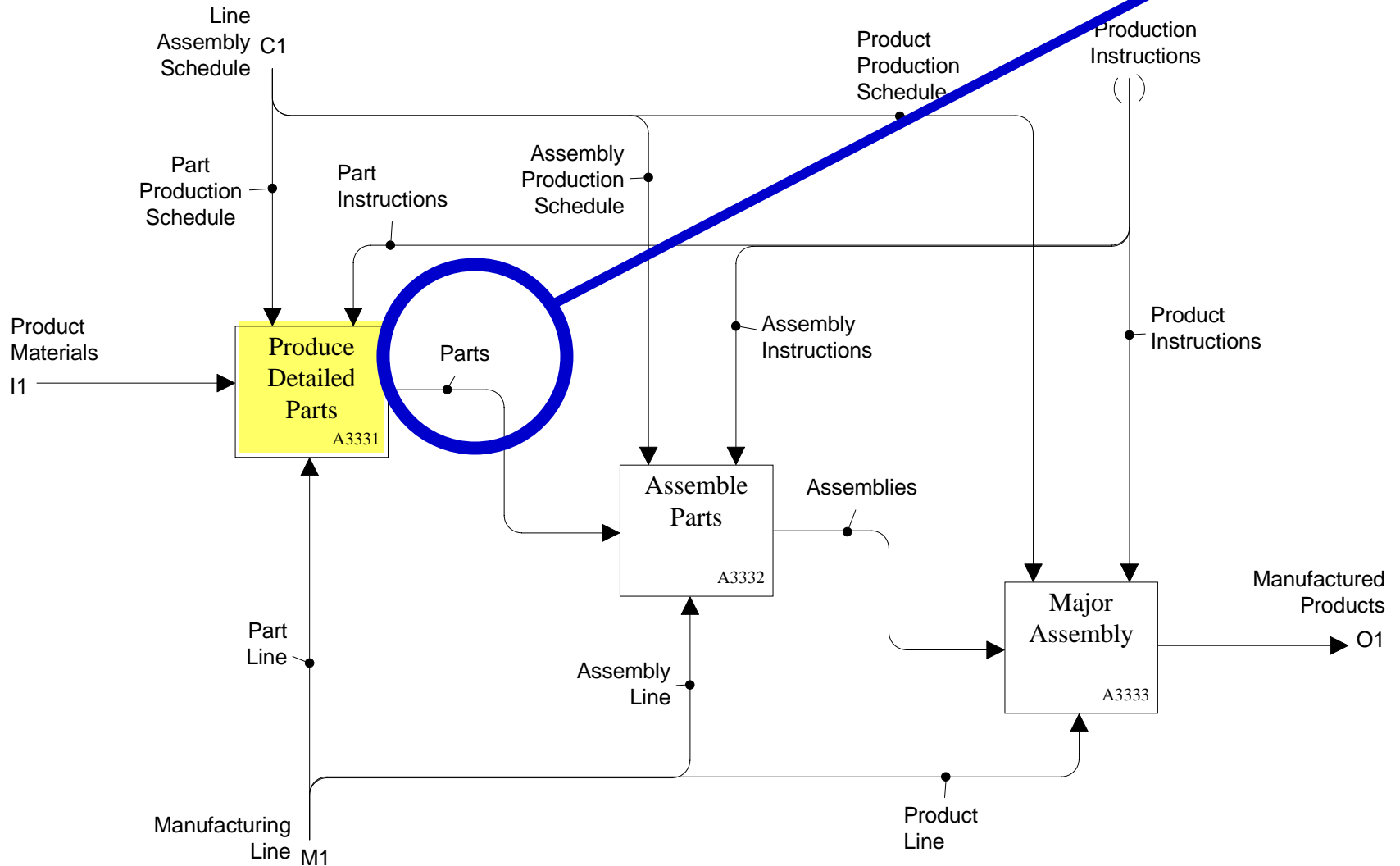


#6 Technical Design Links to the Infrastructure

Define keyword for each distinct end-user action, needed business data, policy, instruction, etc.

- **One Web-based GUI for each end-user action.**
- **One “Live” File for each data, policy, instruction, etc.**
- **One Hyperlink to each Web-based GUI and Live File.**
- **Software Engineering principles used:**
 - Graphical User Interface Design of Web-based Forms
 - Structured Programming (of Office Suite Applications)

★ Technical Design Process Generates Live Data



★ Technical Design Hyperlink to Live Data

The screenshot shows a Microsoft Internet Explorer browser window displaying a web application. The address bar shows the path C:\Process\NCSE\A3331x7.htm. The main content area features a hierarchical menu structure with the following items:

- Company XYZ
- Supplier ABC
- 0. Company XYZ
- 3. Product Mfg and Distribution
- 3.3. Manufacture Product
- 3.3.3. Produce Product
- 3.3.3.1. Produce Detailed Parts

The path from 3.3.3.1. Produce Detailed Parts down to the 'Parts Per Day' link is highlighted with a yellow border. The 'Parts Per Day' link is underlined and blue, indicating it is a hyperlink. A callout box on the right lists the types of links that can be used:

Hyperlink:

- External Website
- Intranet Page
- Web Form
- Application
- Live File

The browser's taskbar at the bottom shows the 'My Computer' icon.

☆ Hyperlink Semantics

Assign different colors to inputs, outputs, controls and mechanisms. Customize at run-time.

- **Novice Users do not customize their browser**
 - They use the multi-colored links to learn the process
- **Infrequent Users keep process and data cues visually present**
 - Override with just one color -- thereby creating reminders
- **Expert Users read process descriptions as ordinary text**
 - Turn off all link colors and turn on the hover color

☆ Conclusion

Software engineering principles are applicable to business-to-business electronic connections.

□ Design before Implementation

- The business partnership is designed correctly

□ Rapid Application Development

- The connection is built fast enough

□ Highly Modular Architecture

- The operations can quickly adapt to changing business direction