The Modeling Activity

Chapter 6

What is a Model?

- A "model" refers to a representation or depiction of various aspects of a WebApp, including its content, functionality, architecture, components, interfaces, navigation, and aesthetics.

- This representation is created using a combination of text, graphics, and diagrams.

- The purpose of creating such a model is to provide a clear and unambiguous way to convey the design and specifications of the WebApp, which can be prone to ambiguity and lack of clarity when described solely in natural language

What model should we build

(ie: Is there a need to build model for everything)

- The best way to identify areas where better understanding (through modeling) is required is to think about the combination of two things:

  - **The process being followed**

    (and, hence, the *viewpoints* that are relevant to the activities, actions, and tasks being carried out), and

  - **The product being developed**

    the *things* that need to be understood and modeled.

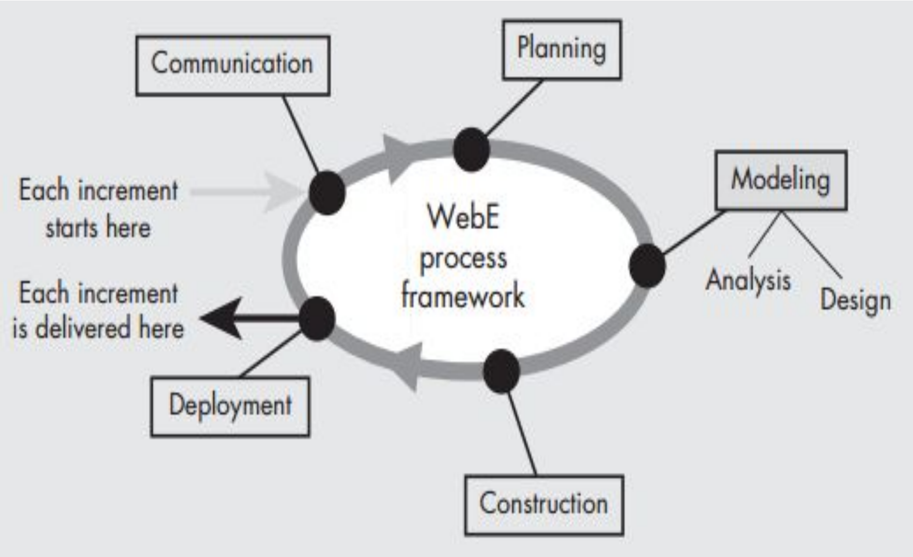
Modeling

- Analysis modeling

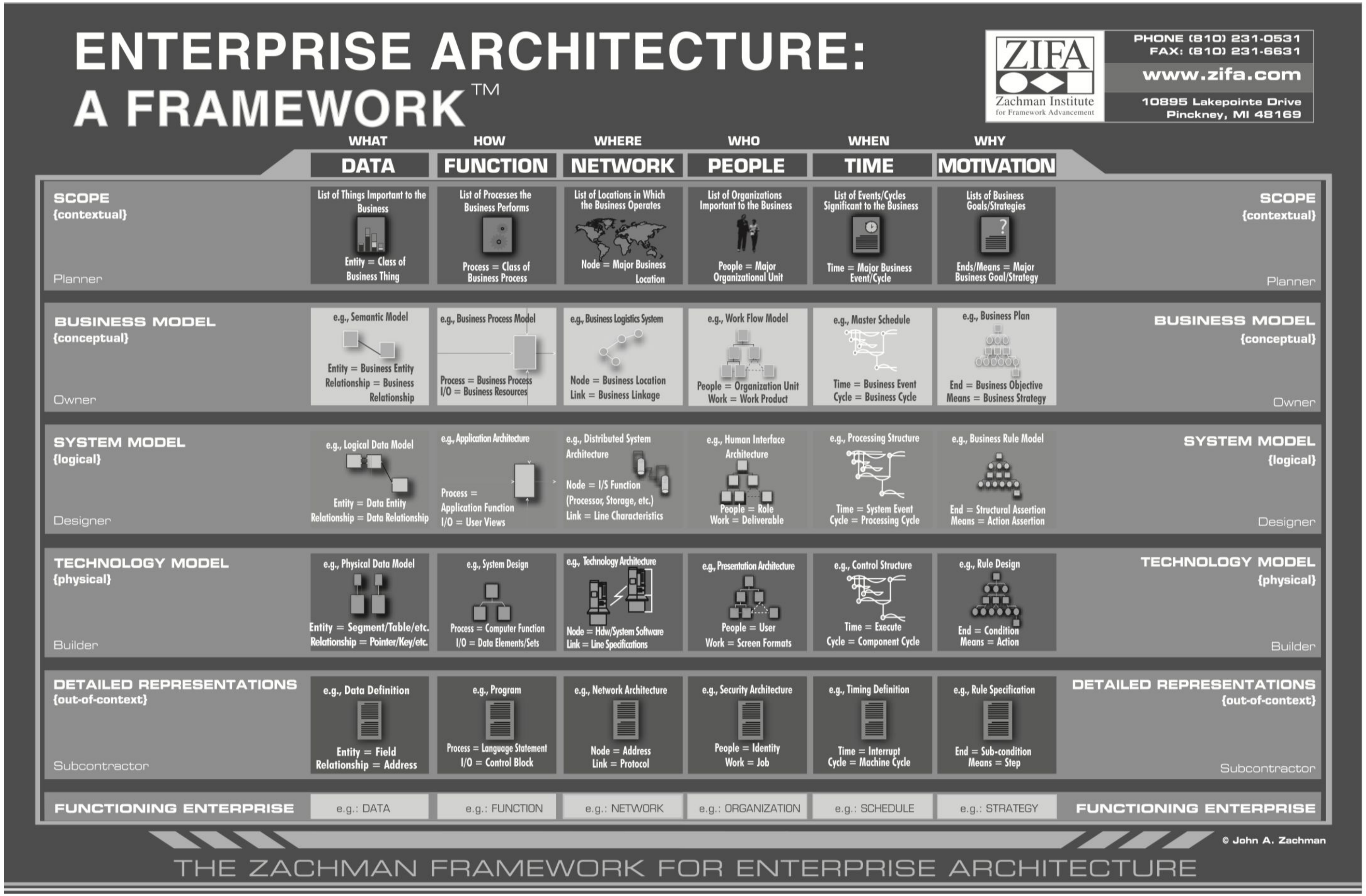
… to understand the nature of the problem being addressed and the “shape” of the WebApp that will allow you to address that problem

- Design modeling

… to understanding the internal structure of the WebApp being developed and how this creates the shape of the WebApp that was identified by the analysis model



Zackman's Modeling Framework



ENTERPRISE ARCHITECTURE: A FRAMEWORK™

ZIFA
Zachman Institute
for Framework Advancement

PHONE (810) 231-0531
FAX: (810) 231-6631
www.zifa.com
10895 Lakepointe Drive
Pinckney, MI 48169

	WHAT	HOW	WHERE	WHO	WHEN	WHY	
	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION	
SCOPE {contextual} Planner	List of Things Important to the Business Entity = Class of Business Thing	List of Processes the Business Performs Process = Class of Business Process	List of Locations in Which the Business Operates Node = Major Business Location	List of Organizations Important to the Business People = Major Organizational Unit	List of Events/Cycles Significant to the Business Time = Major Business Event/Cycle	Lists of Business Goals/Strategies Ends/Means = Major Business Goal/Strategy	SCOPE {contextual} Planner
BUSINESS MODEL {conceptual} Owner	e.g., Semantic Model Entity = Business Entity Relationship = Business Relationship	e.g., Business Process Model Process = Business Process I/O = Business Resources	e.g., Business Logistics System Node = Business Location Link = Business Linkage	e.g., Work Flow Model People = Organization Unit Work = Work Product	e.g., Master Schedule Time = Business Event Cycle = Business Cycle	e.g., Business Plan End = Business Objective Means = Business Strategy	BUSINESS MODEL {conceptual} Owner
SYSTEM MODEL {logical} Designer	e.g., Logical Data Model Entity = Data Entity Relationship = Data Relationship	e.g., Application Architecture Process = Application Function I/O = User Views	e.g., Distributed System Architecture Node = I/S Function (Processor, Storage, etc.) Link = Line Characteristics	e.g., Human Interface Architecture People = Role Work = Deliverable	e.g., Processing Structure Time = System Event Cycle = Processing Cycle	e.g., Business Rule Model End = Structural Assertion Means = Action Assertion	SYSTEM MODEL {logical} Designer
TECHNOLOGY MODEL {physical} Builder	e.g., Physical Data Model Entity = Segment/Table/etc. Relationship = Pointer/Key/etc.	e.g., System Design Process = Computer Function I/O = Data Elements/Sets	e.g., Technology Architecture Node = Hdw/System Software Link = Line Specifications	e.g., Presentation Architecture People = User Work = Screen Formats	e.g., Control Structure Time = Execute Cycle = Component Cycle	e.g., Rule Design End = Condition Means = Action	TECHNOLOGY MODEL {physical} Builder
DETAILED REPRESENTATIONS {out-of-context} Subcontractor	e.g., Data Definition Entity = Field Relationship = Address	e.g., Program Process = Language Statement I/O = Control Block	e.g., Network Architecture Node = Address Link = Protocol	e.g., Security Architecture People = Identity Work = Job	e.g., Timing Definition Time = Interrupt Cycle = Machine Cycle	e.g., Rule Specification End = Sub-condition Means = Step	DETAILED REPRESENTATIONS {out-of-context} Subcontractor
FUNCTIONING ENTERPRISE	e.g.: DATA	e.g.: FUNCTION	e.g.: NETWORK	e.g.: ORGANIZATION	e.g.: SCHEDULE	e.g.: STRATEGY	FUNCTIONING ENTERPRISE

© John A. Zachman

THE ZACHMAN FRAMEWORK FOR ENTERPRISE ARCHITECTURE

Web Application Architecture Framework (WAAF)

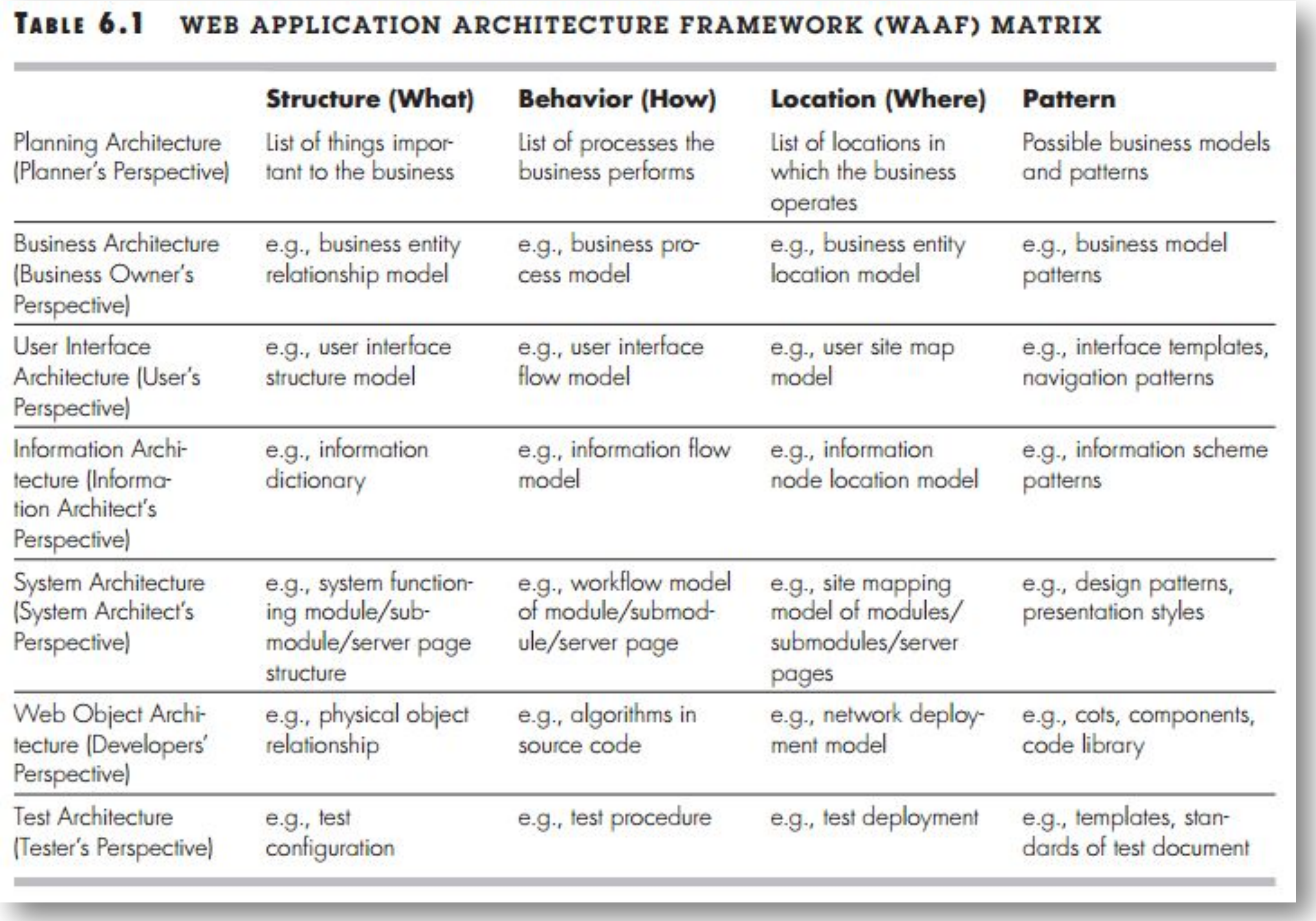


TABLE 6.1 WEB APPLICATION ARCHITECTURE FRAMEWORK (WAAF) MATRIX

	Structure (What)	**Behavior (How)**	**Location (Where)**	**Pattern**
Planning Architecture (Planner's Perspective)	List of things important to the business	List of processes the business performs	List of locations in which the business operates	Possible business models and patterns
Business Architecture (Business Owner's Perspective)	e.g., business entity relationship model	e.g., business process model	e.g., business entity location model	e.g., business model patterns
User Interface Architecture (User's Perspective)	e.g., user interface structure model	e.g., user interface flow model	e.g., user site map model	e.g., interface templates, navigation patterns
Information Architecture (Information Architect's Perspective)	e.g., information dictionary	e.g., information flow model	e.g., information node location model	e.g., information scheme patterns
System Architecture (System Architect's Perspective)	e.g., system functioning module/submodule/server page structure	e.g., workflow model of module/submodule/server page	e.g., site mapping model of modules/submodules/server pages	e.g., design patterns, presentation styles
Web Object Architecture (Developers' Perspective)	e.g., physical object relationship	e.g., algorithms in source code	e.g., network deployment model	e.g., cots, components, code library
Test Architecture (Tester's Perspective)	e.g., test configuration	e.g., test procedure	e.g., test deployment	e.g., templates, standards of test document

Web Application Architecture Framework (WAAF)

Classifies concerns related to development of WebApp along two dimensions.

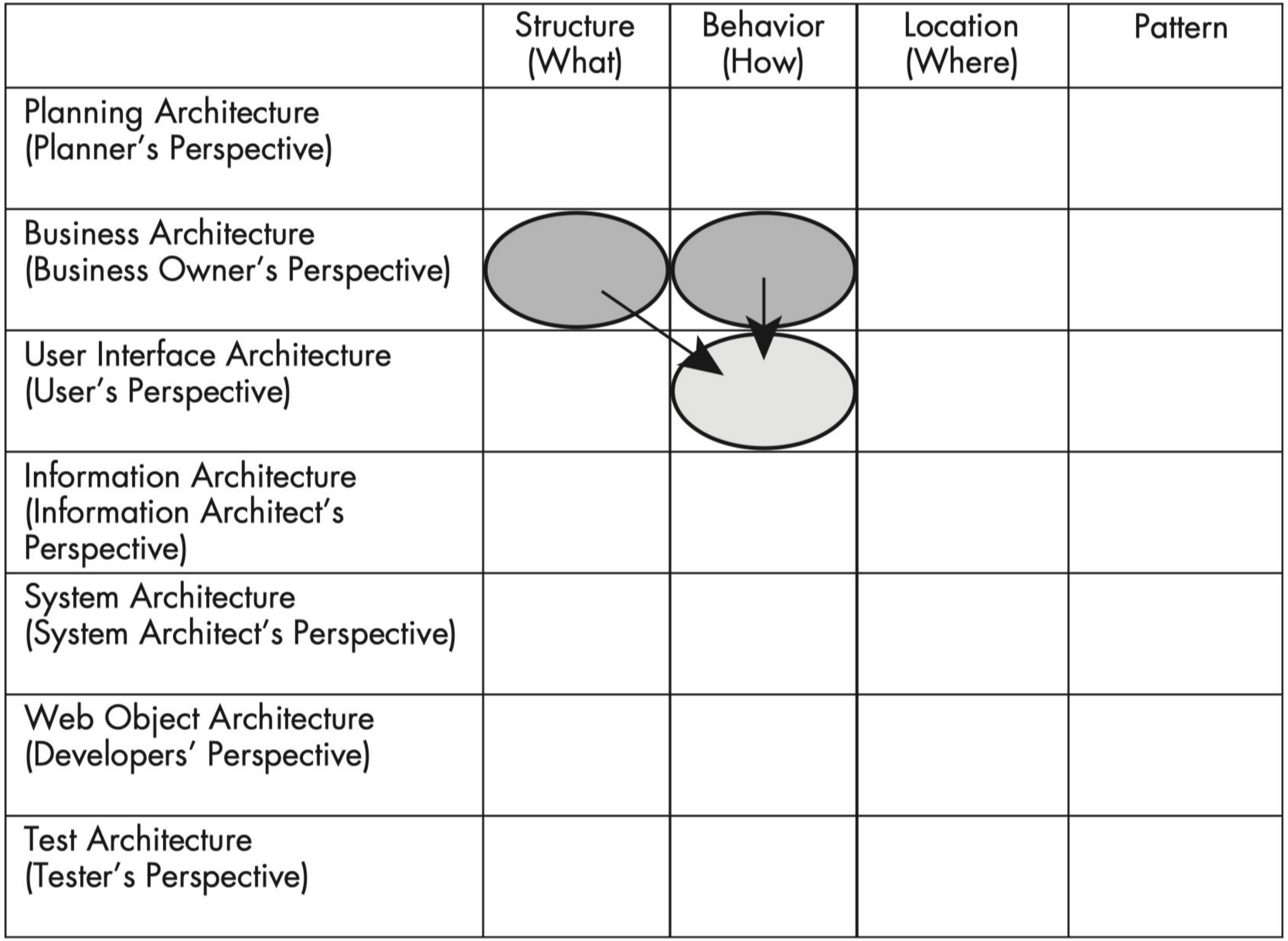
- **The horizontal dimension** presents the perspective of the different participants in the WebE process: business owners, WebApp users, information architects, application architects, developers, and testers.

- **The vertical dimension** classifies the architectures into four categories that capture the main modeling domains: structure (*what*), behavior (*how*), location (*where*),

Web Application Architecture Framework (WAAF)

FIGURE 6.3

WAAF models associated with the analysis task, *Create an interaction model for complex scenarios.*

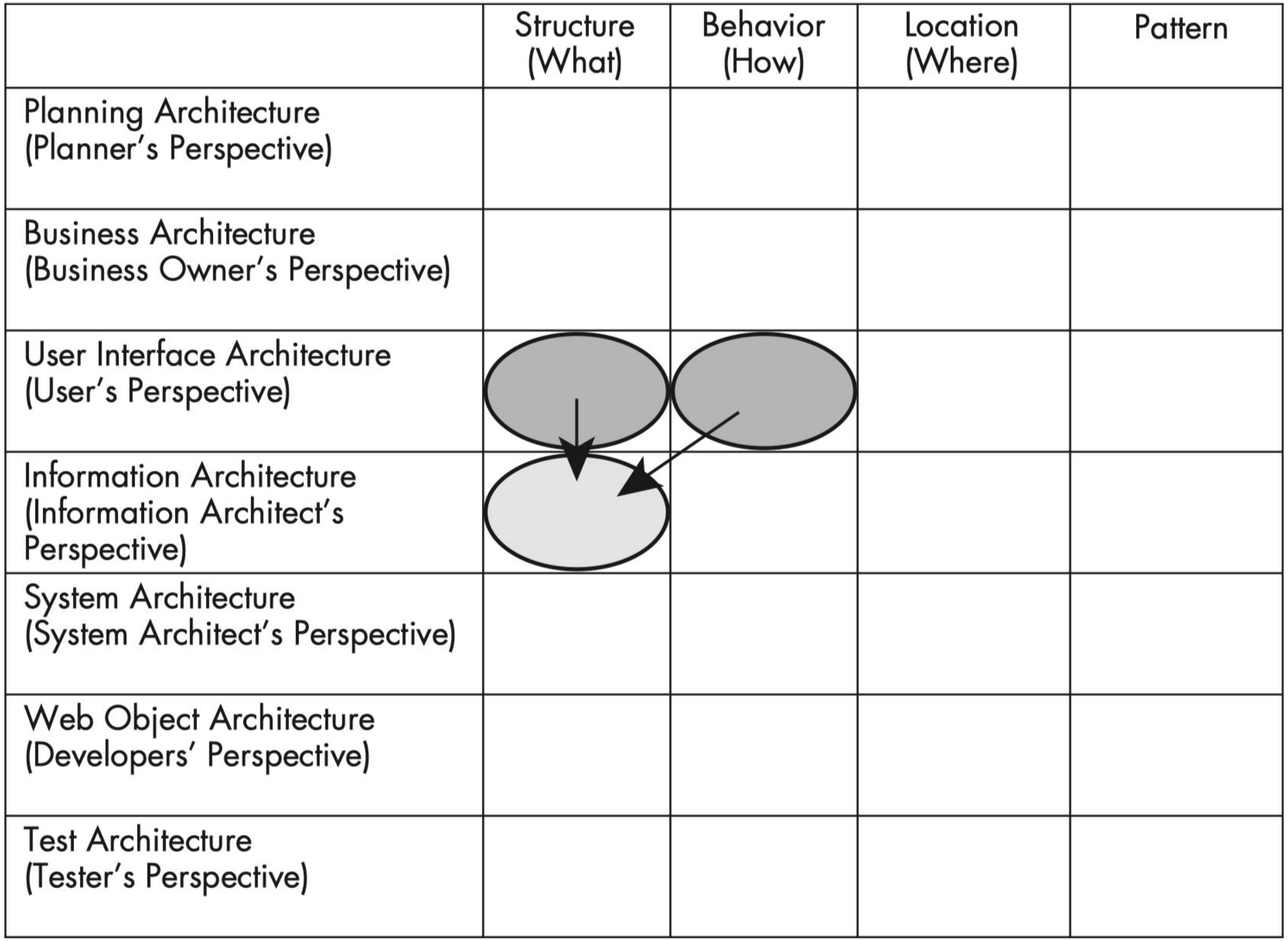


	Structure (What)	Behavior (How)	Location (Where)	Pattern
Planning Architecture (Planner's Perspective)				
Business Architecture (Business Owner's Perspective)	●	●		
User Interface Architecture (User's Perspective)		○		
Information Architecture (Information Architect's Perspective)				
System Architecture (System Architect's Perspective)				
Web Object Architecture (Developers' Perspective)				
Test Architecture (Tester's Perspective)				

Web Application Architecture Framework (WAAF)

FIGURE 6.4

WAAF models associated with design task, *Design the navigation scheme.*



	Structure (What)	Behavior (How)	Location (Where)	Pattern
Planning Architecture (Planner's Perspective)				
Business Architecture (Business Owner's Perspective)				
User Interface Architecture (User's Perspective)				
Information Architecture (Information Architect's Perspective)				
System Architecture (System Architect's Perspective)				
Web Object Architecture (Developers' Perspective)				
Test Architecture (Tester's Perspective)				

Modeling Language

- In the context of Web engineering, a modeling language incorporates a set of notations, terms, and/or symbols, as well as the rules for establishing associations between them.

Modeling language should provide the following capabilities to Model **Functionality**

- Ability to model integration and connectivity.
- Ability to support pattern modeling.
- Ability to represent concepts in a technology-neutral fashion.
- Ability to model sophisticated system functionality.

Modeling language should provide the following capabilities to Model **Information Content**

- Ability to model presentation-level concepts

- Ability to model navigational structure and behavior

- Ability to model user interactions with the information

- Ability to model user roles and user groups

- Ability to model content.

Modeling language should provide the following **Generic Capabilities**:

- Ability to model business domain concepts
- Ability to link business models with the technical architecture
- Ability to link information with functionality.
- Ability to maintain system integrity
- Ability to support understanding and communication
- Ability to support Web system life cycle management.