



Primitives and Design Patterns for Top-Down SOA Implementations

Michael zur Muehlen

Stevens Institute of Technology

Hoboken NJ



HOWE SCHOOL
of Technology Management

STEVENS
Institute of Technology



Engineering vs. Architecture

- ▶ If you show the same circuit diagram to two electrical engineers they will be able to understand the diagram because
 - The symbols are drawn the same way everywhere
 - The symbols have a well-defined meaning (semantics)
 - All electrical engineers are trained on the same set of symbols

Resistor symbol

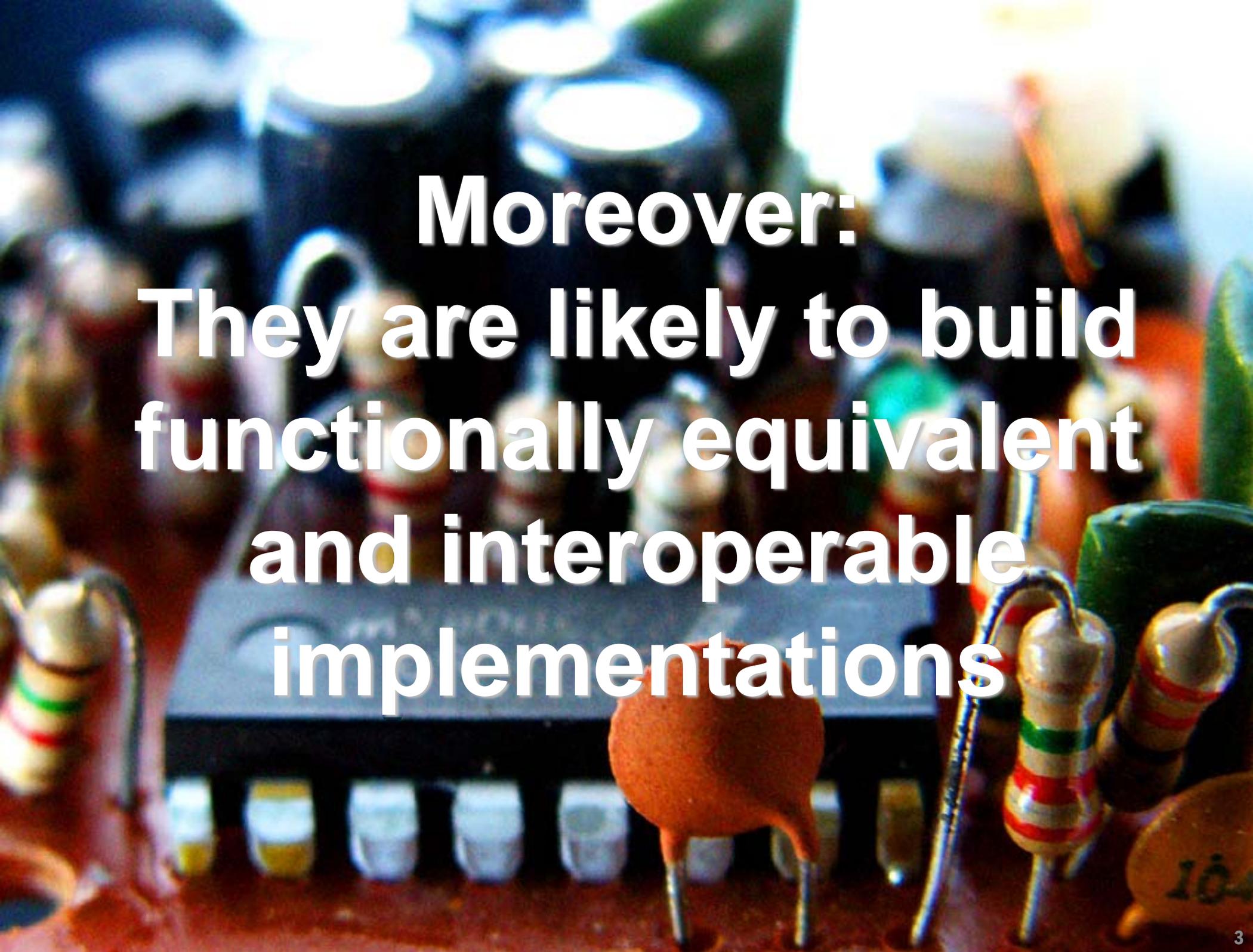


Capacitor symbol



This agreed upon representation of electrical engineering allows a common understanding...





**Moreover:
They are likely to build
functionally equivalent
and interoperable
implementations**

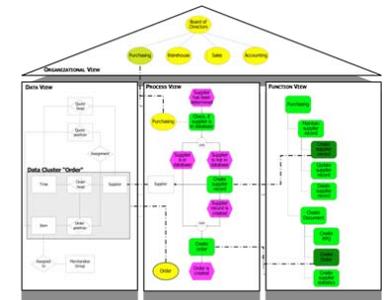
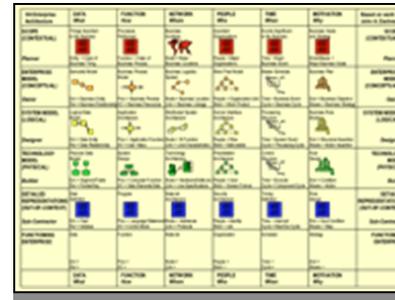


Enterprise Architecture

▶ Many Frameworks



	Function View	Information View	Resource View	Organization View
Requirements Definition	<ul style="list-style-type: none"> Domains Domain Processes Business Processes Enterprise Activities Events 	<ul style="list-style-type: none"> Enterprise Objects Object Views Object Relationships Information Elements Integrity Rules 	<ul style="list-style-type: none"> Capabilities 	<ul style="list-style-type: none"> Responsibility Authority
Design Specifications	<ul style="list-style-type: none"> Specified Functional Operations 	<ul style="list-style-type: none"> External Schemata Conceptual Schemata Integrity Constraints Database Transactions 	<ul style="list-style-type: none"> Specified Capabilities Specified Resources Specified Resource Units 	<ul style="list-style-type: none"> Organization Units Organization Cells
Implementation Descriptions	<ul style="list-style-type: none"> Implemented Functional Operations 	<ul style="list-style-type: none"> Implemented External Schemata Internal Schema Logical Data Schema Physical Data Schema 	<ul style="list-style-type: none"> Implemented Capabilities Implemented Resources Implemented Resource Units 	<ul style="list-style-type: none"> Implemented Organization Units Implemented Organization Cells



▶ Many Views

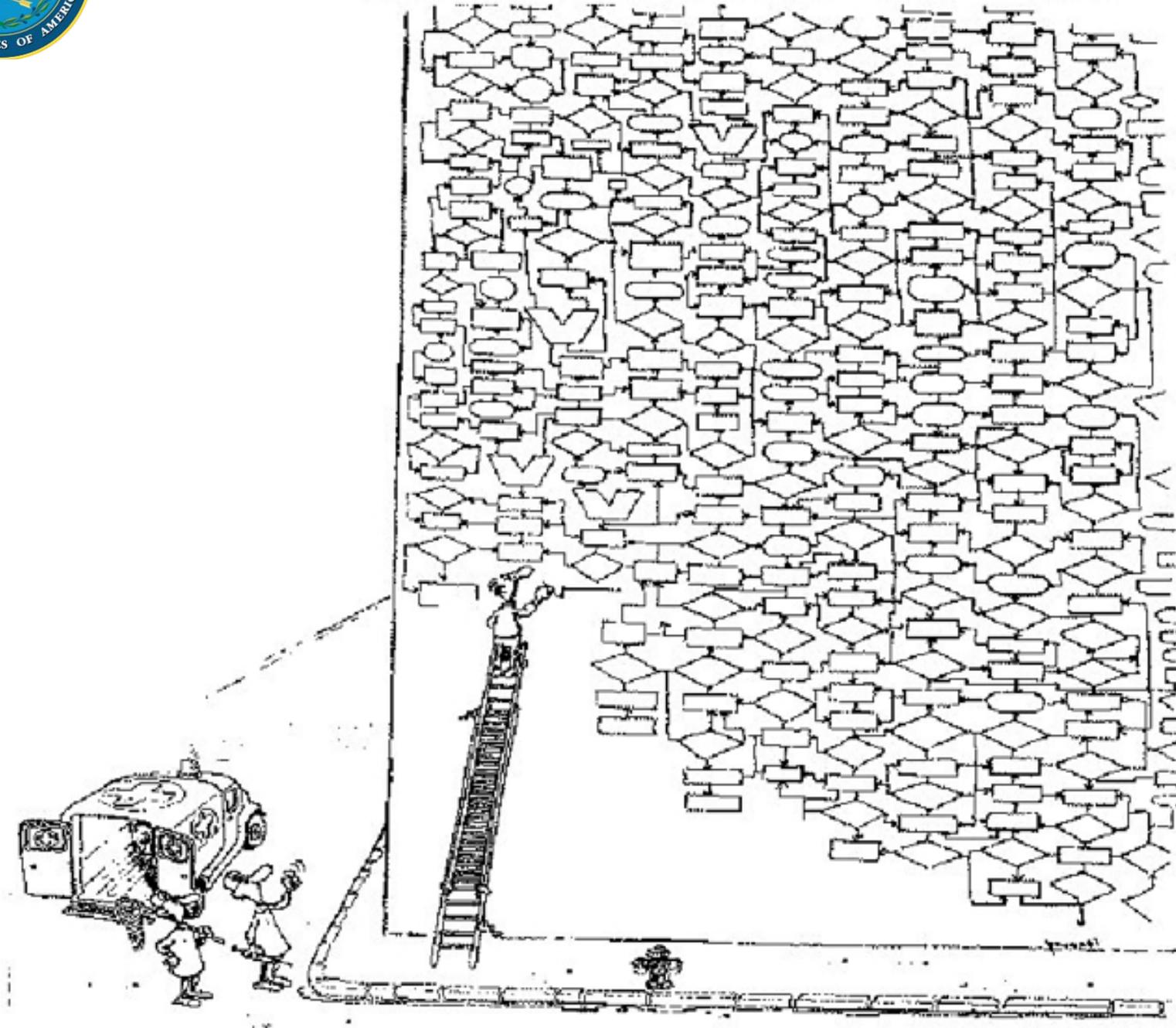


▶ Many Techniques

- UML, IDEF, BPMN, RAD, EPC, PowerPoint and many, many others...



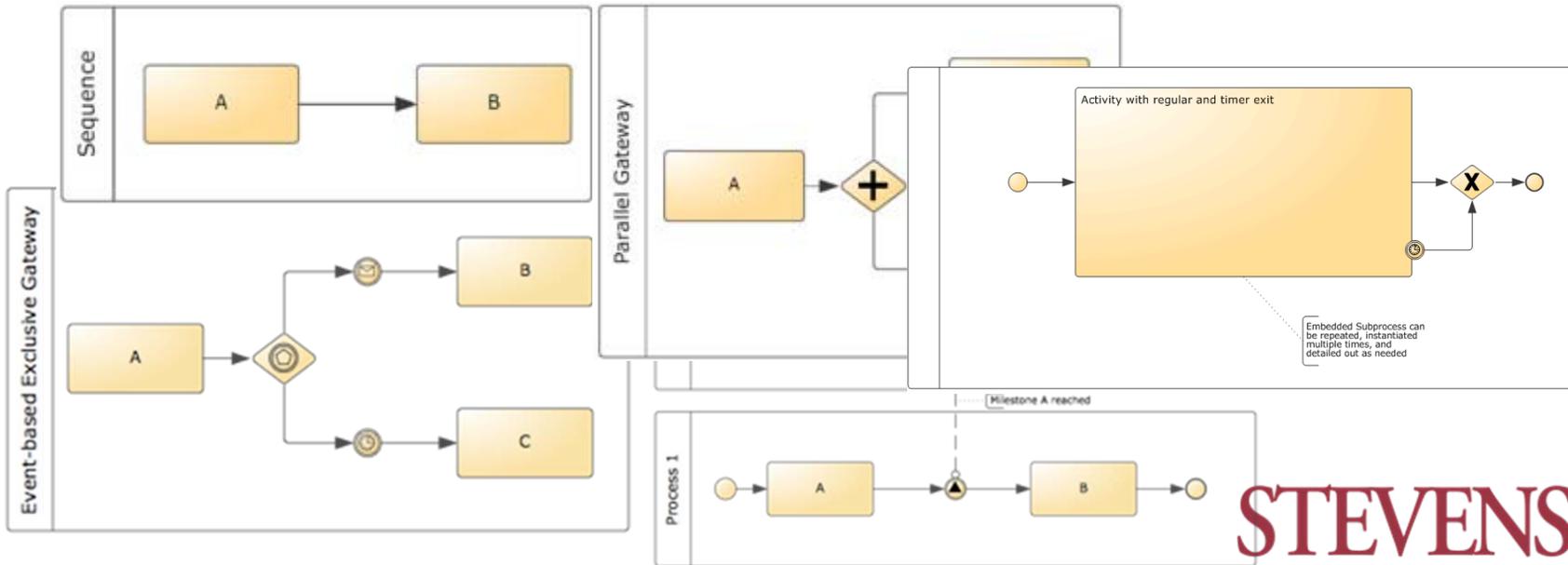
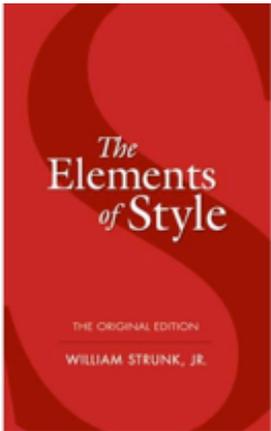
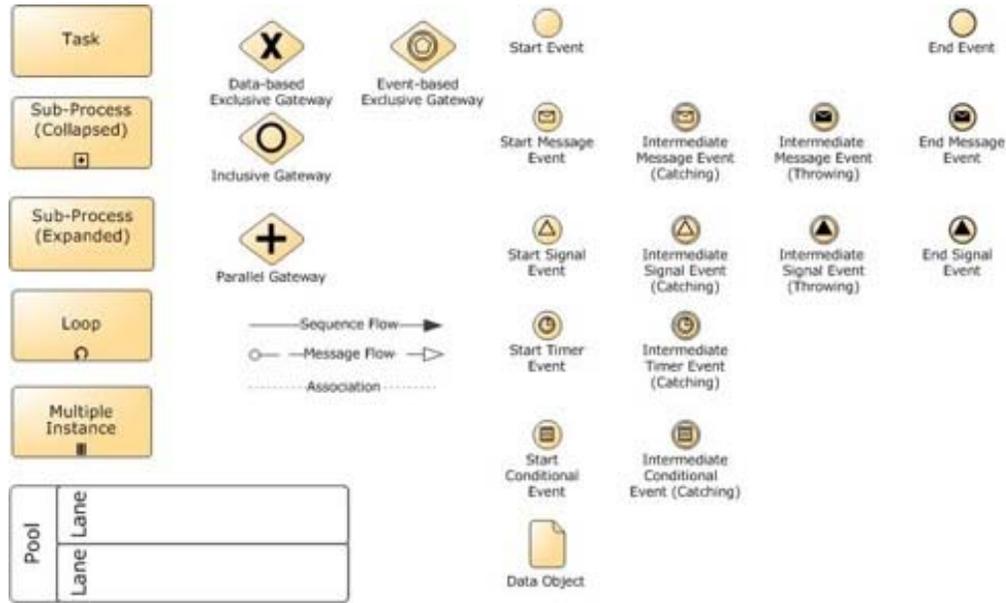
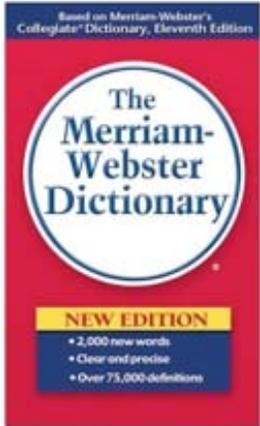
Modeling Freedom



STEVENS
Institute of Technology

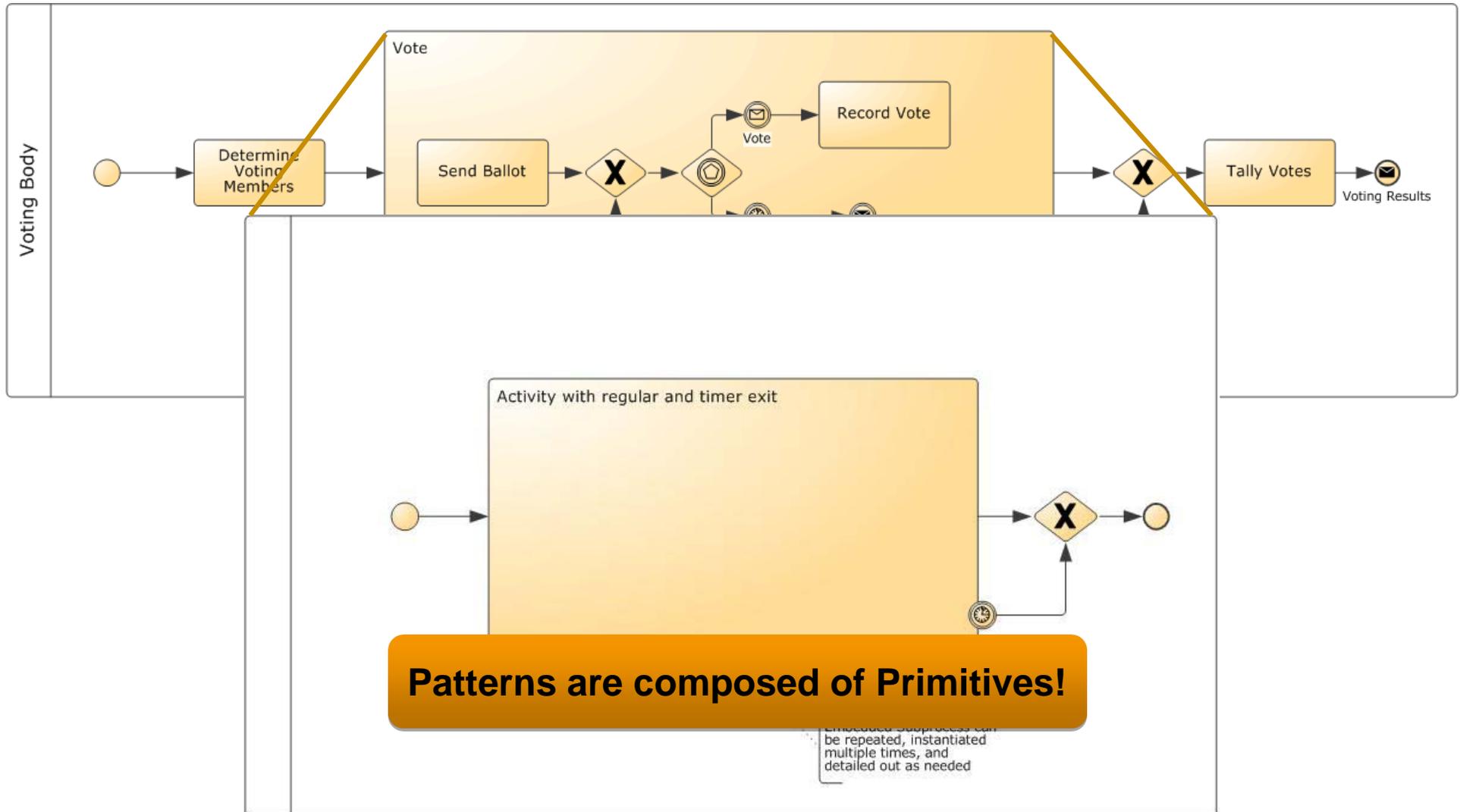


Design Primitives and Patterns



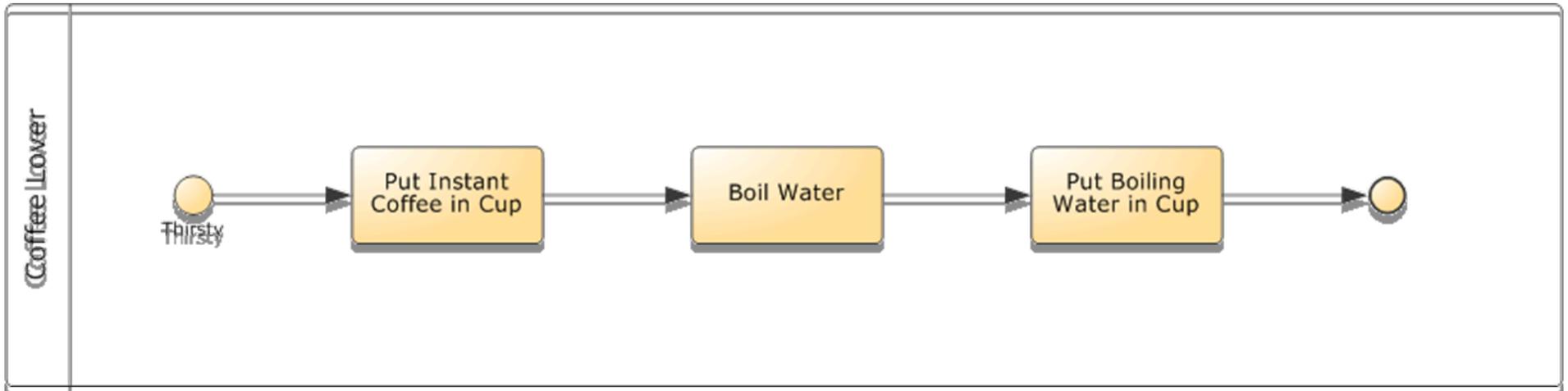


Low- and High-Level Patterns



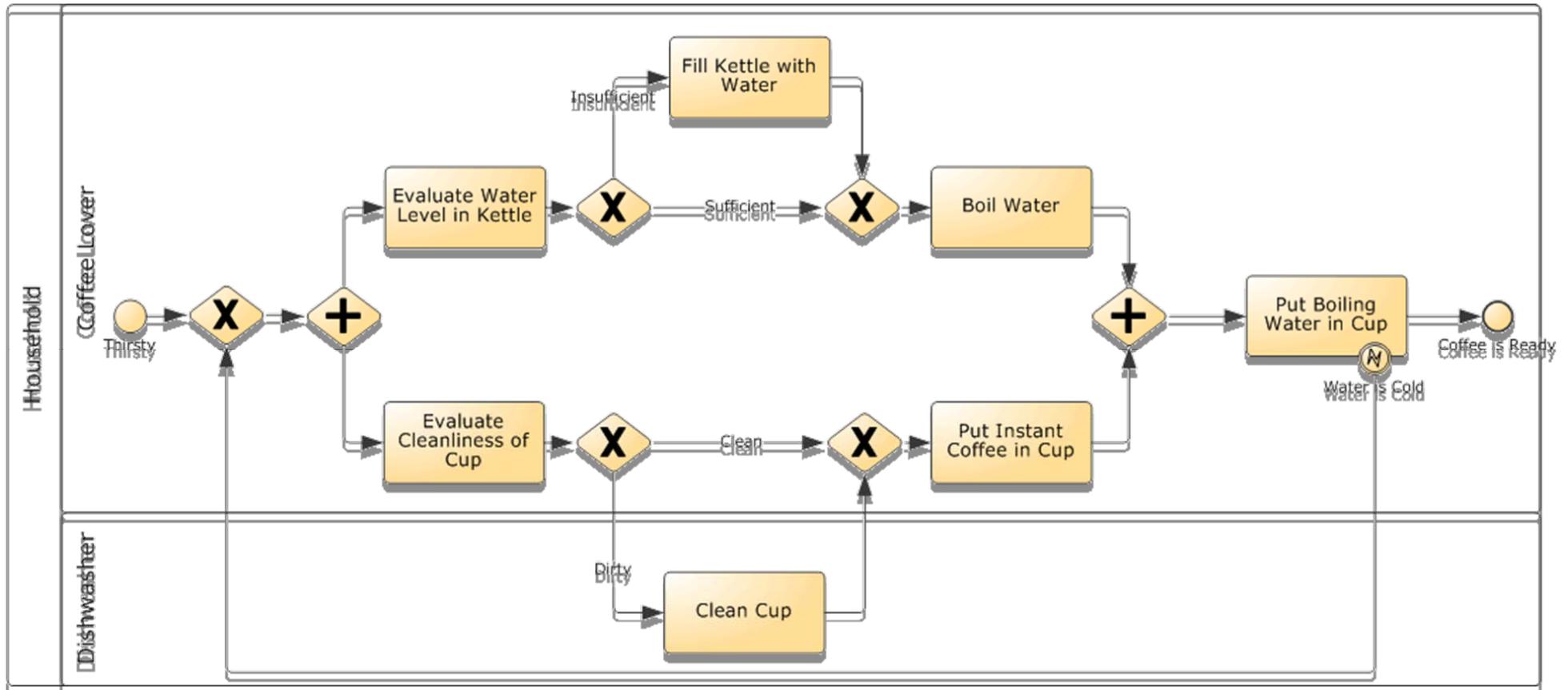


The Nescafé Process



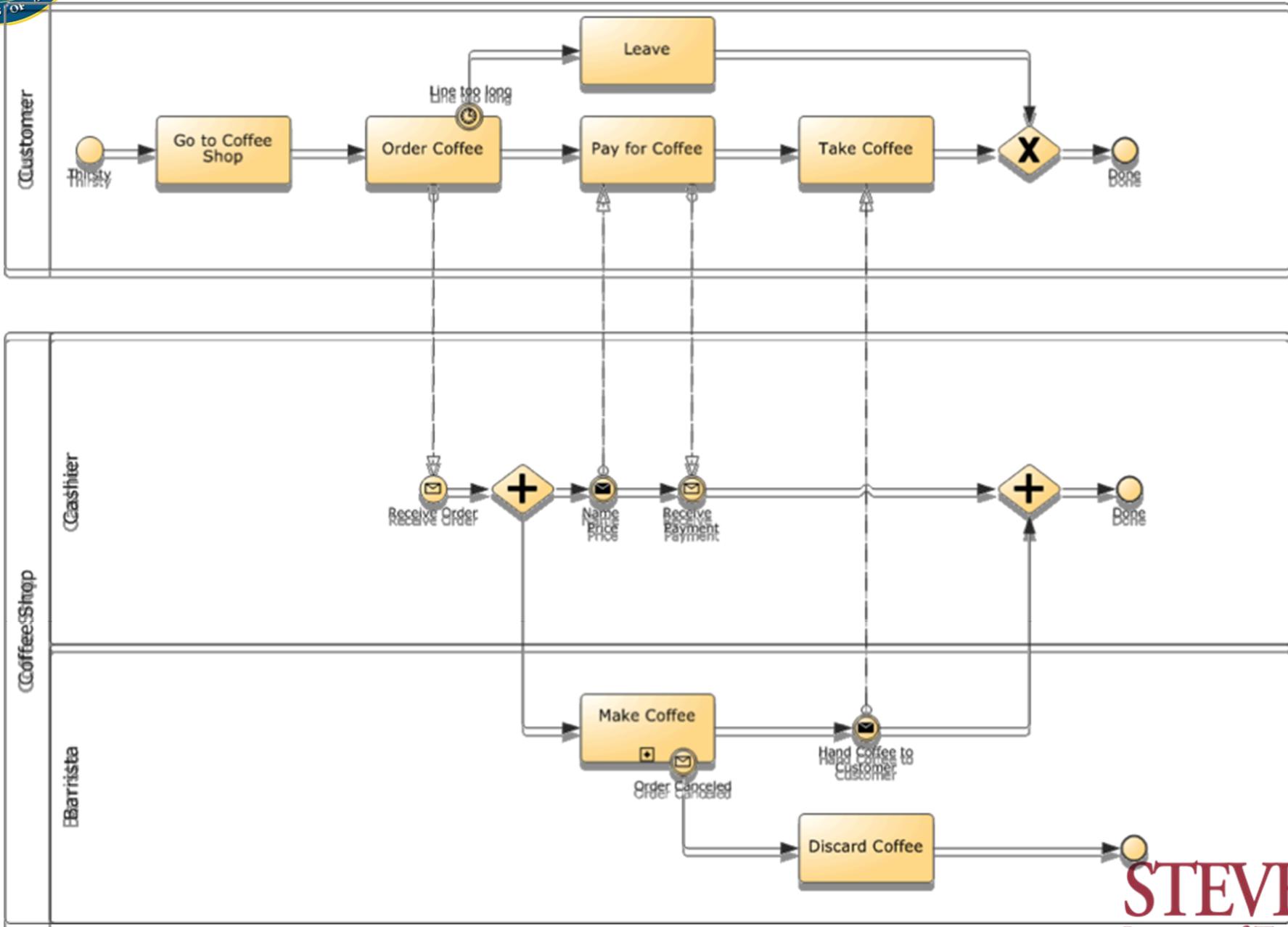


The Espresso Machine Process



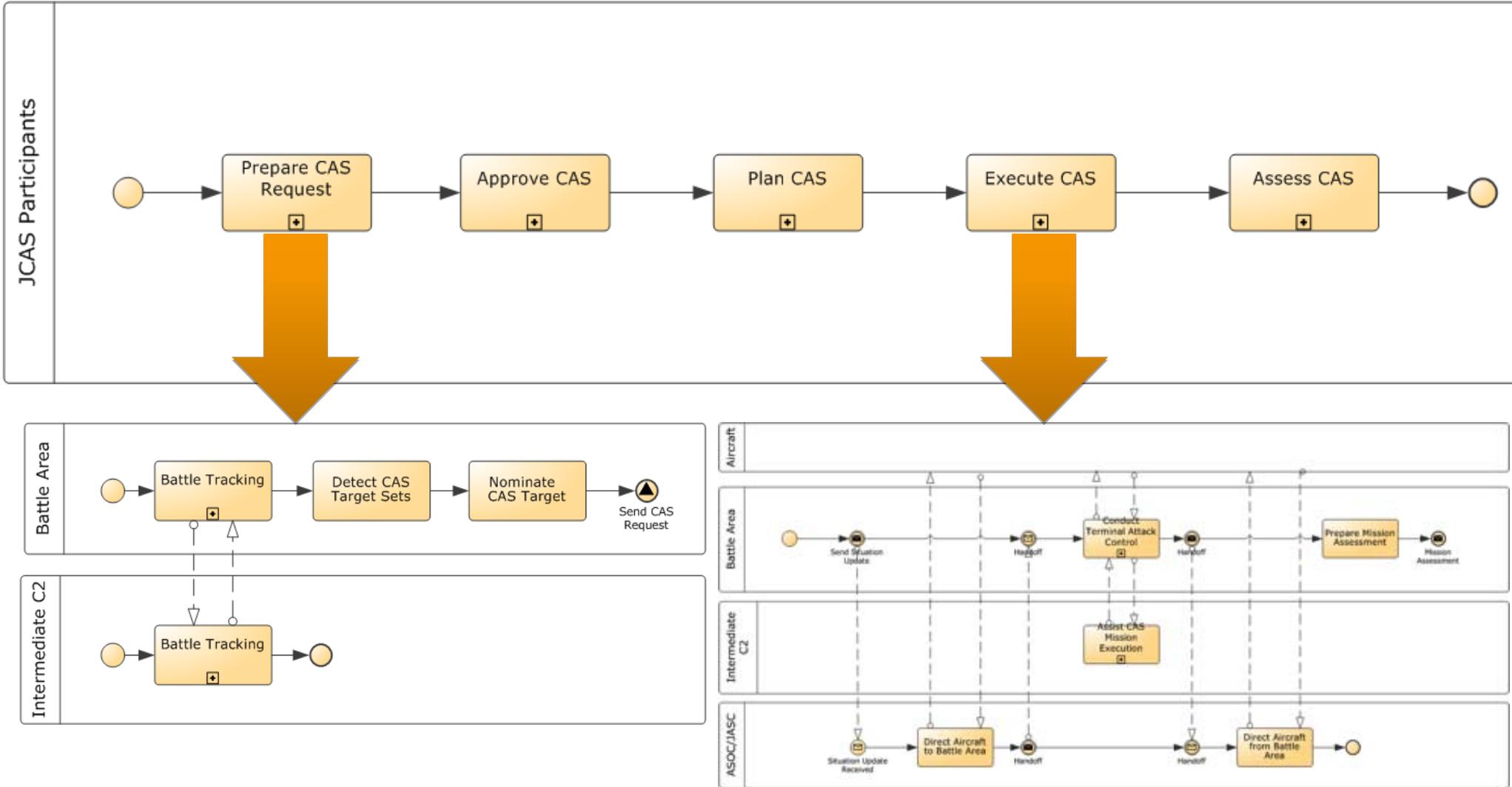


The Starbucks Process



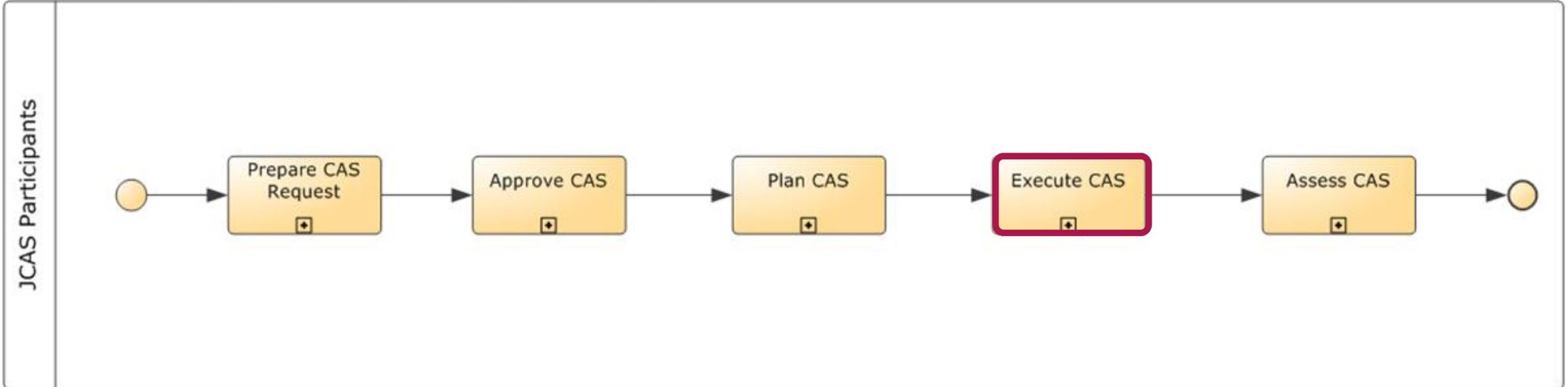


Systematic Composition



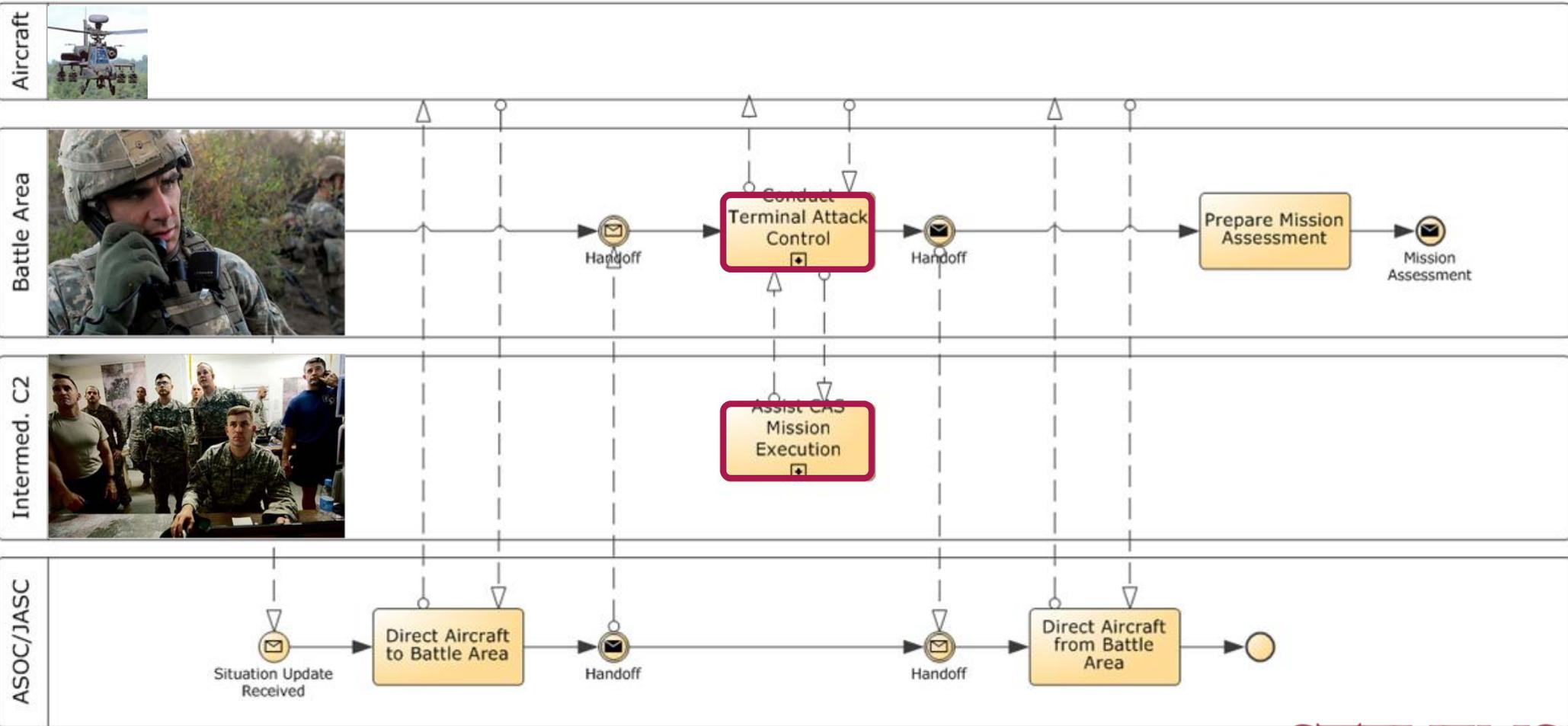


3-Level Hierarchy: Milestones





3-Level Hierarchy: Collaboration

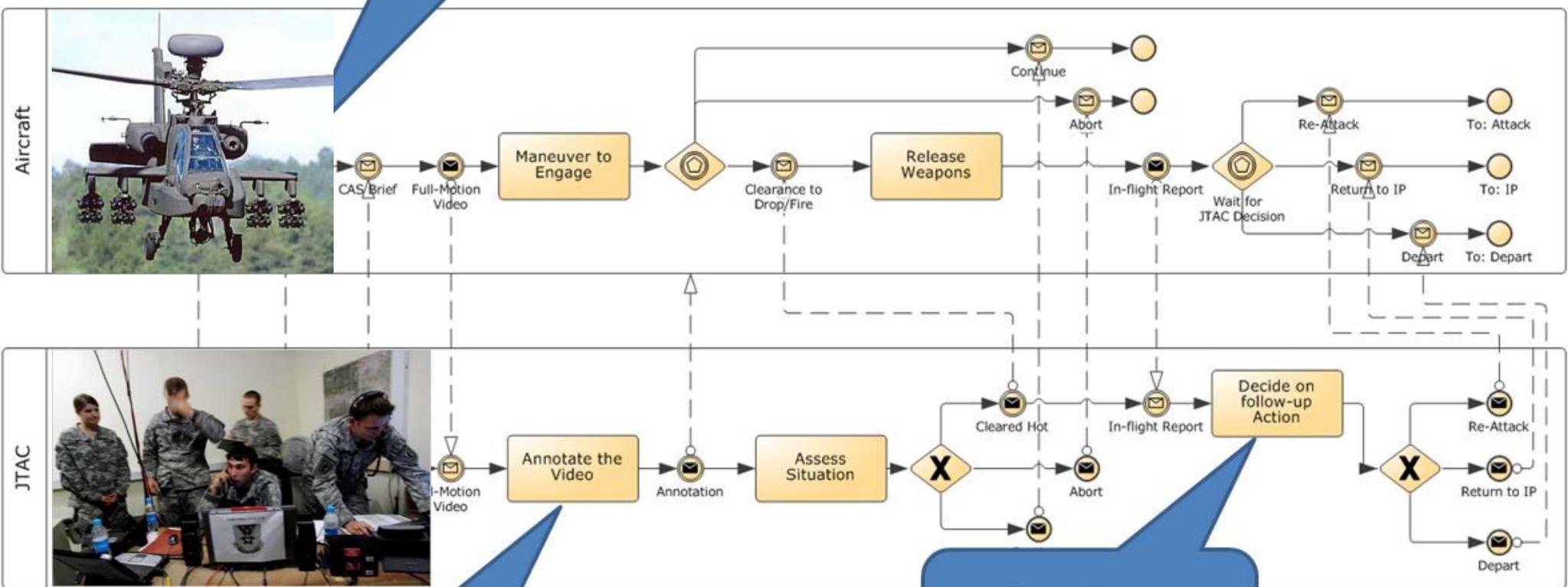




3-Level Hierarchy: Procedures



Focus on detailed message exchange

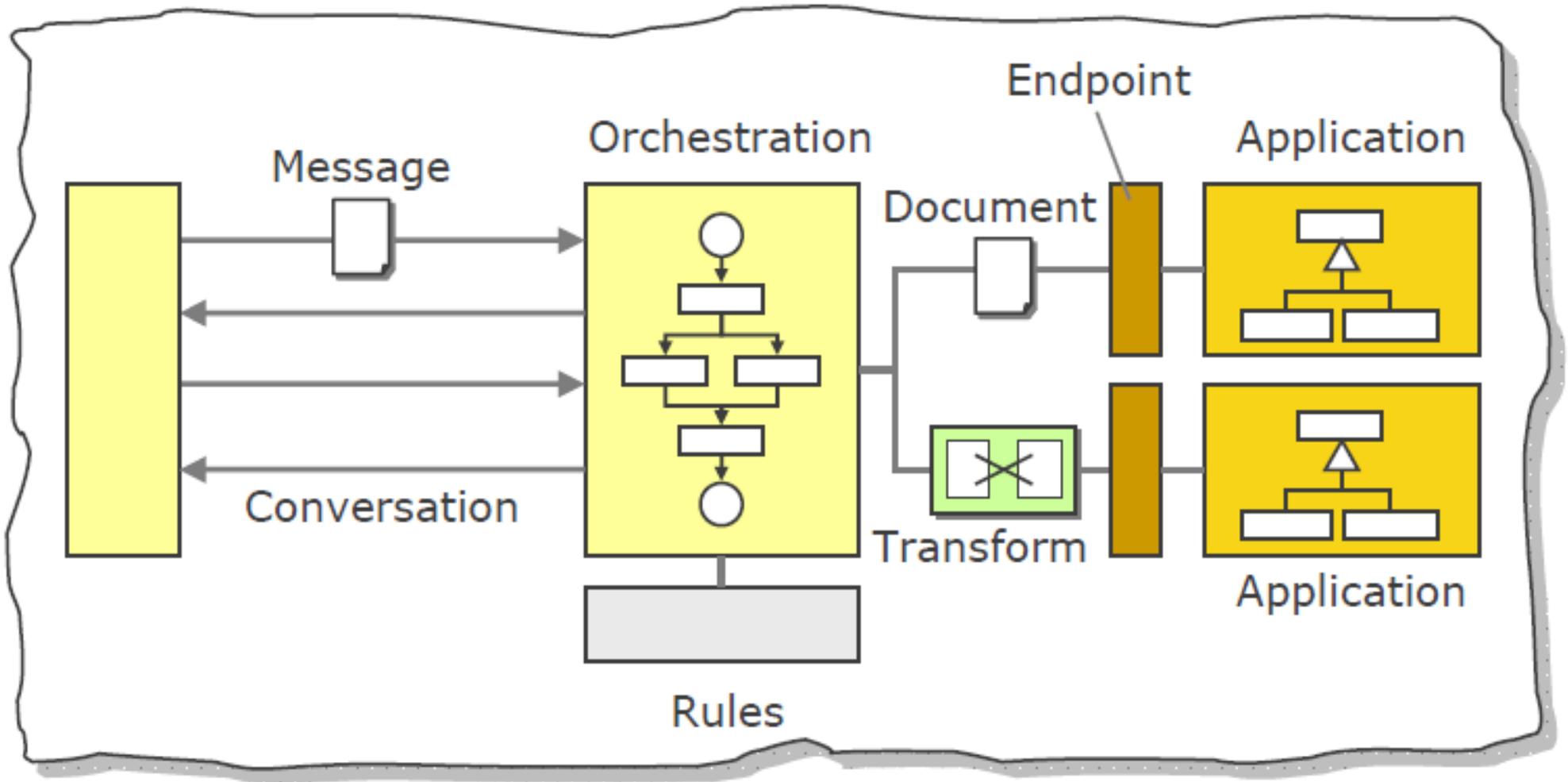


Actionable activity descriptions

Explicit decision-making activities



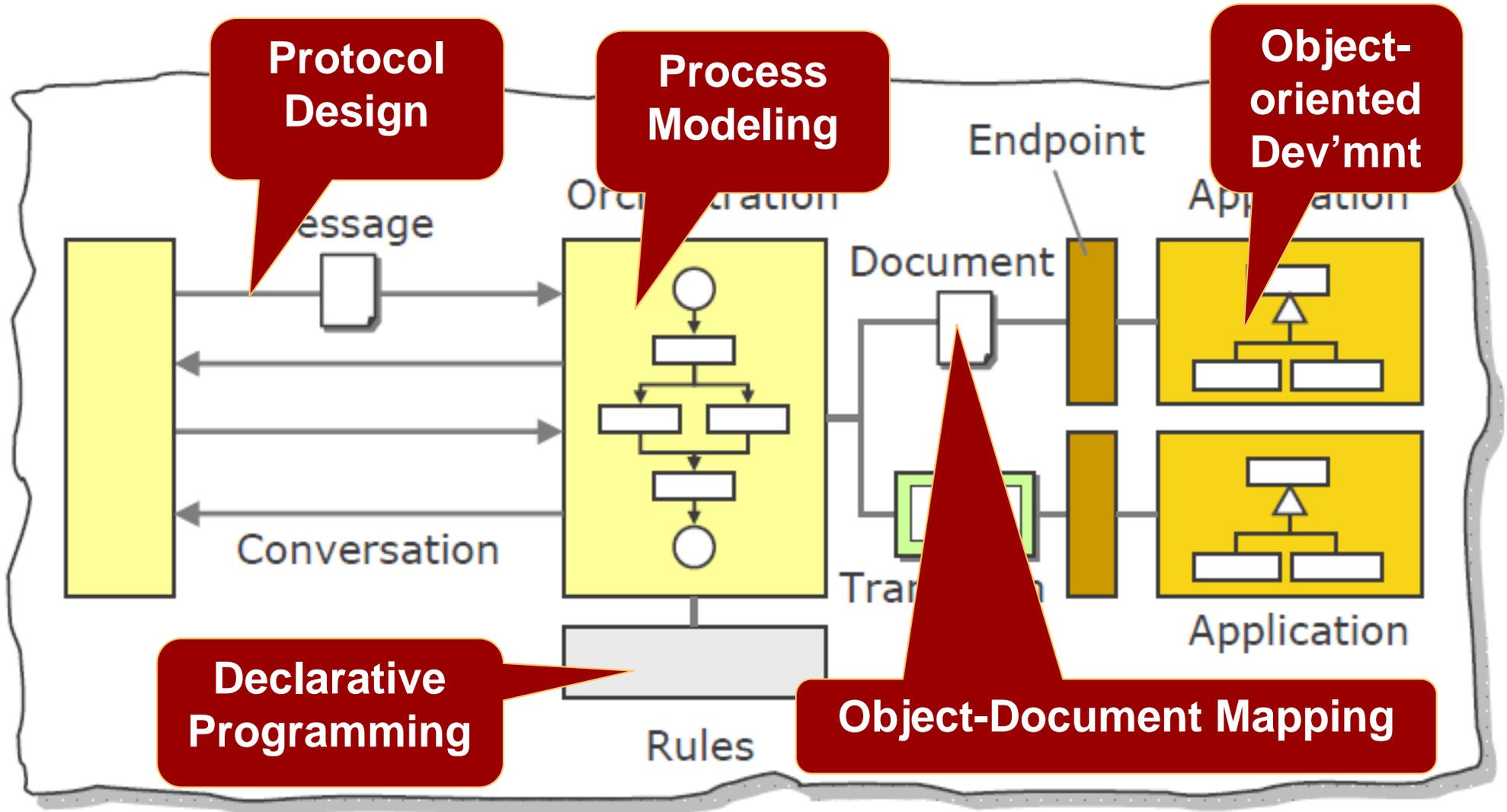
Hohpe's SOA on a Napkin



Source: Gregor Hohpe (2009)



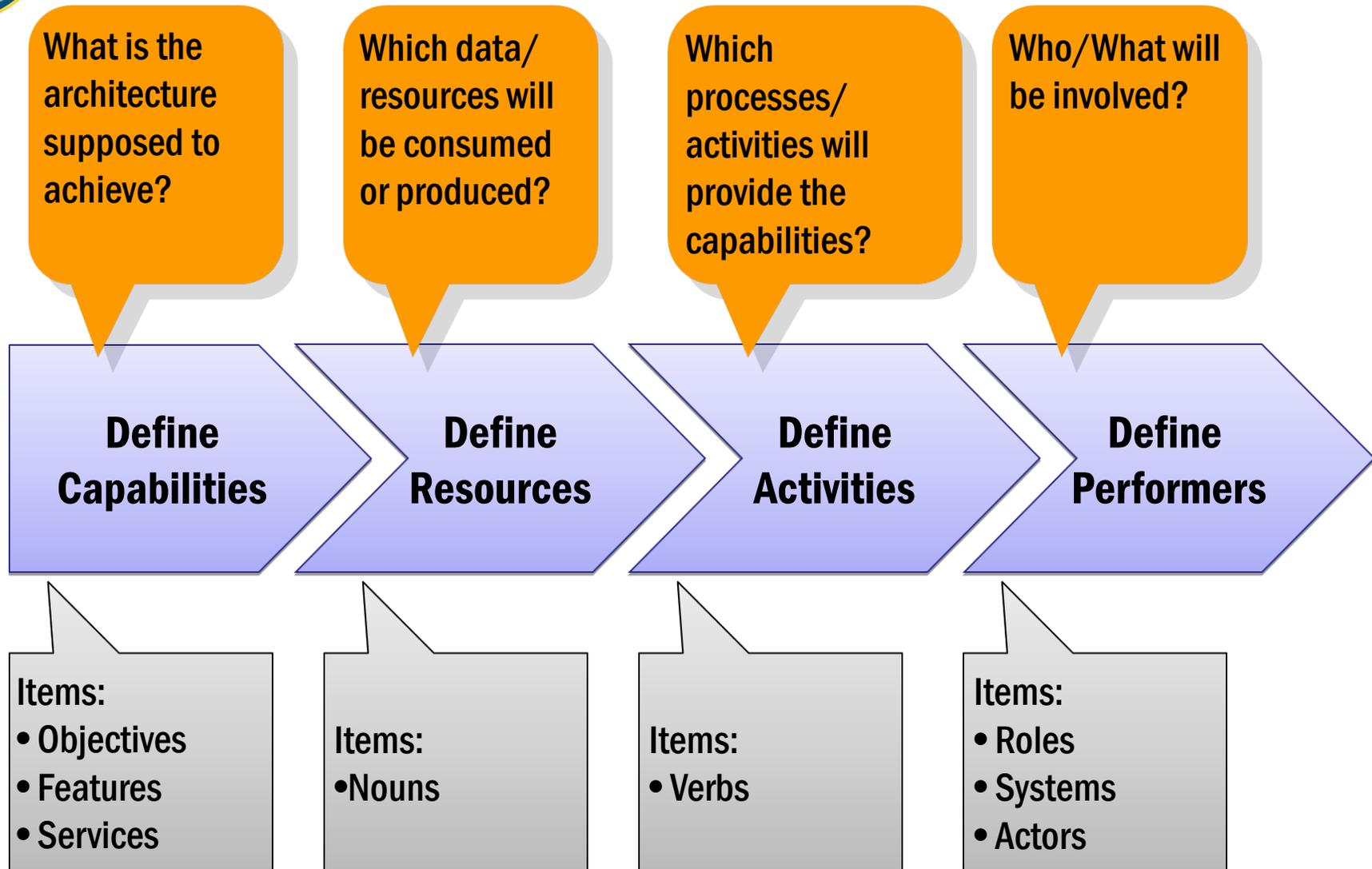
Hohpe's SOA on a Napkin



Source: Gregor Hohpe (2009)

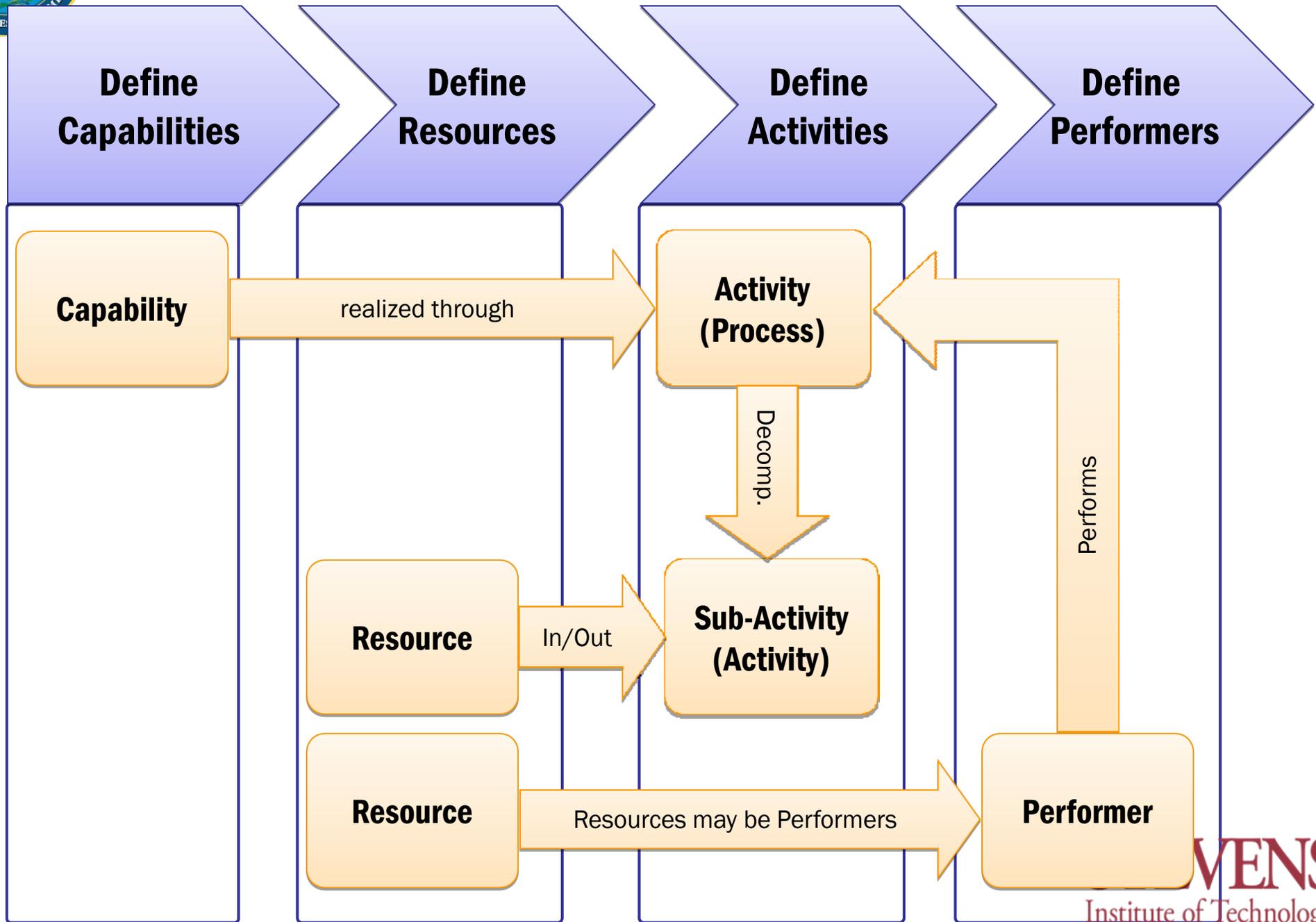


Define Your Vocabulary First





How the Vocabulary Relates





Summary

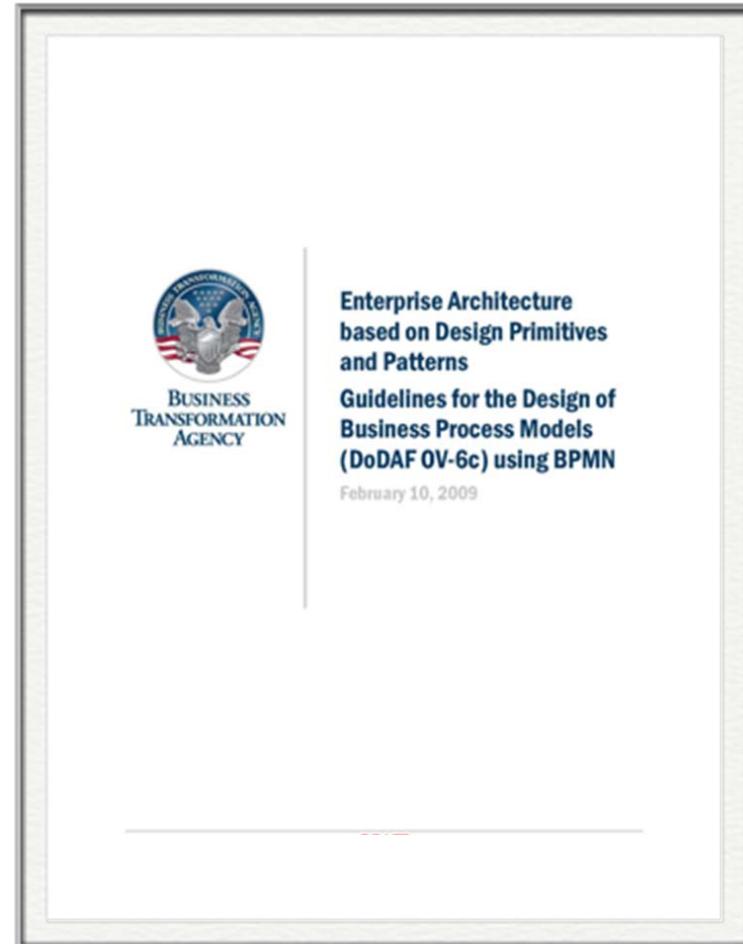
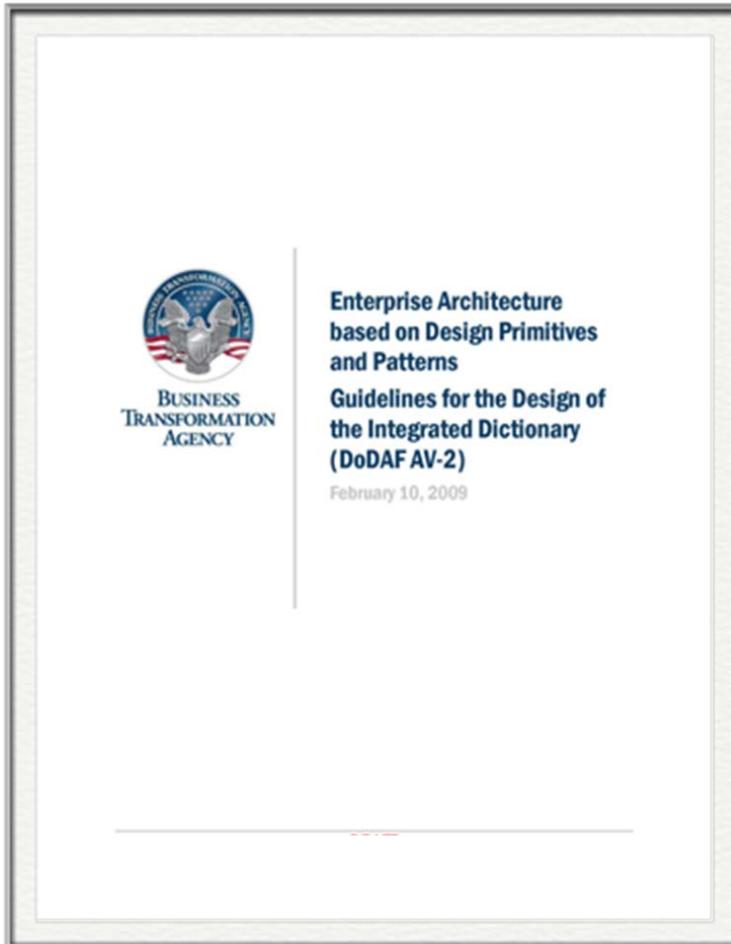
- ▶ Minimize Diagram Types: Fewer primitives to learn
- ▶ Standardize Patterns: Unified use of modeling language
- ▶ Define Vocabulary first: Lexicon ties model types together

- ▶ **A realistic design standard for modelers, architects, and tool vendors**



For More Information

Available on DKO:



Email: Michael.zurMuehlen@stevens.edu