Chapter IV
EA Knowledge for ACE Deployment

ABSTRACT

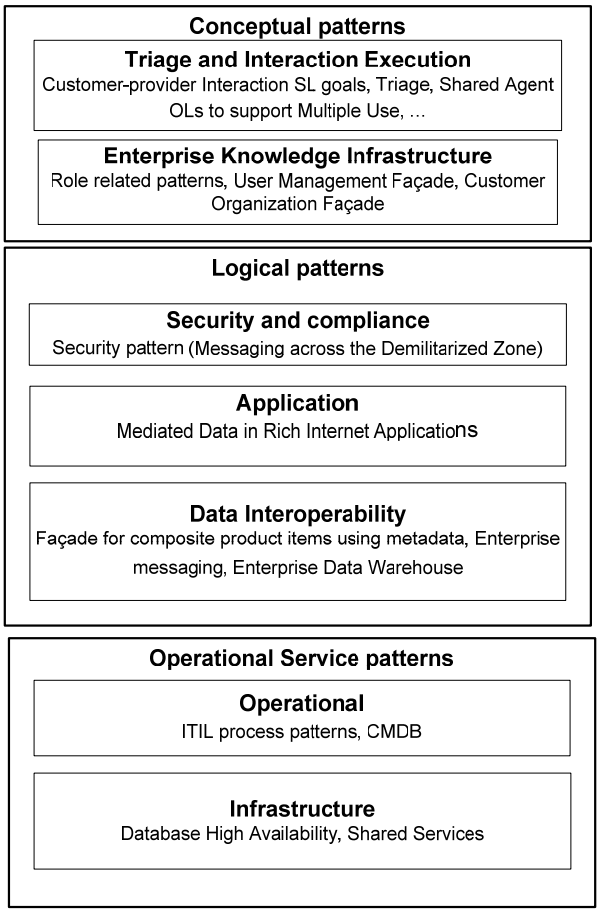
The knowledge infrastructure for enterprise architecture presented here has a taxonomy of useful patterns and pattern applications illustrated in Figure 1. The applications help deploy EKI and enable operations as illustrated herein.

What architecture patterns inform the EA team?

- What are the patterns for enabling Interaction goals with existing enterprise systems?
- What are the logical patterns enabling enterprise integration?
- How are underlying enterprise systems and tools enlisted within these patterns?

As emphasized previously, since the organization progresses towards more comprehensive or customer circumstance-based services, it becomes necessary to support many new types of Requests. From the business perspective, the underlying enterprise interoperability problem is typically stated as a requirement to produce an improved business result from services implemented in software tools. These include communication endpoints by which the systems can be considered to be components that will interact with each other and thereby form a new, integrated service capable of performing new functions. In turn, each system that contributes

Figure 1. Taxonomy of Enterprise Architecture Patterns



information or functionality is often required to expose new services. Thus, the trend is toward exposing more-and-more functionality from existing applications and using interoperability to *compose* these functions in different and new ways.

Related work in Software Product Line 2008 is relevant as the goal here is also to create a base of reusable knowledge. According to this Software Engineering Institute site:

A Software Product Line (SPL) is a set of software-intensive systems that share a common, managed set of features satisfying the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way. Product line adoption involves moving from some form of developing software-intensive systems with a single-system mentality to developing them