

# Ontologies to Complement Software Architectures

Kevin Bodie

Pitney Bowes Inc.

An ontology is a description of the terminology, concepts, and the relationships among those concepts relevant to a particular domain.

Consider the following:

- Semantics of published APIs
- Component models
- Frameworks
- Architecture specifications

To give software architecture consistent meaning across an organization and then to external organizations a common understanding of the architectural concept model must be developed. Concepts may include business objects, services, components, infrastructure, etc. While these concepts may provide a common language for the architects, often their meaning can be different in engineering organizations, the business development arena, the C-Suite, marketing, etc.

In developing an Internet Product Architecture Framework (iPAF) it became necessary to create and define views and create a common language. This language was based on many audiences and through a laborious and time consuming process the views were adopted by a diverse group of people from many organizations internal and external.

This process was used several times to:

- Describe published APIs for reusable components.
- Illustrate component models following functional decomposition of a system
- Show the use of frameworks through sample applications as part of developer level documentation to “illustrate by example.”

- Show areas of software reuse in terms of percentage of an application or solution and in terms of number of times reused

Of particular interest is the use of a common language and description of a software system to create an architecture specification, an artifact created as part of our software and systems development process. This architecture specification should address the following:

- Architecturally significant use cases
- Architecture interaction diagrams
- Use of best practices patterns
- Non functional requirements such as performance, scalability, load balancing, etc.
- Extensibility to future product directions
- Extensibility to new product lines sharing common architecture requirements

The architecture specification is reviewed in the early phases of system and software development. Among the goals of the architecture review process is identification of the key risks, issues, development options, best practices used in the system, etc. Once identified these architecturally significant strategies are examined in detail to provide a high level of confidence that the system and software architecture is scalable, well constructed, well designed, suitable for performance and other non-functional requirements, capable of meeting functional requirements, and supportable.

The success of the described architecture specification and review requires a consistent and common understanding of the architectural concept model. This includes both terminology and illustration of the model. My ideas on the model include but are limited to:

- Component view (component decomposition showing reuse across products)
- Solutions view (products interacting together to create a solution)

- Product view (product offering composed of components and application logic)
- Workflow or business process view (mapping of business process into solutions or product views)
- Description of the application program interface (API)
- Role of off the shelf technologies, middleware, database, device level, and infrastructure level components
- Interfaces at all levels
- Layers, how many there are and what they are called: solutions, business logic, middleware, data access, communications, etc.