“The 3+1 Views of Architecture (in 3D)”:  
An Amplification of the 4+1 Viewpoint Framework

Mark Kennaley  
Fourth Medium Consulting Inc.

Abstract

Processes and techniques for the development of enterprise software are best served when infused with practical experience. Pragmatic reflection on what works and what doesn’t is critical to enable successful adoption of any process. One aspect of developing software intensive systems is that of architecture description. A very popular approach to software architecture description is the 4+1 Views of Architecture [1]. From practical experience applying this framework, this paper discusses an amplification of the 4+1 Views approach - the “3+1 Views of Architecture – in 3D”. It is also compliant with the best practices described within the international standard IEEE 1471 – Recommended Practice for Architectural Description of Software Intensive Systems [2].

1. Introduction

Software architecture description (AD) deals with articulating the significant design decisions of a software system to those who care about it – namely the stakeholders of the system being developed. An AD typically is achieved through leveraging visual modeling to communicate in a precise, engineering way. An example of a visual modeling notation suite is the Unified Modeling Language (UML) [3].

Software architecture refers to fundamental or significant design decisions or choices about components and the architectural styles that guide the solution [4]. The design “with a capital-D” decisions are those which could materially affect the value proposition of the product. As such, the types of systems that could materially suffer large-order losses accrue value from the mitigation of such risk. Such risk mitigation is effected through architectural reasoning techniques captured within, among other things, “blueprints”.

One very popular framework for describing the architecture of software intensive systems is the 4+1 Views of Architecture. Figure 1 illustrates this seminal work on architectural description.

![Figure 1. Original 4+1 Views of Architecture approach to describing architecture](image)

From experience on the ground applying this framework for describing software architecture [5], some recurring challenges include: 1) the original 4+1 paper gave limited treatment to the “+1” Scenario view and argued it as being redundant rather than scoping & binding; later within the Unified Process (UP) [6] community, this was somewhat remedied, but lost the intent of dynamics in this view in favor of pushing scenarios/UC Realizations to the Logical View; 2) in practice, rarely are directory structures of implementation source files modeled at anything but a coarse grained level in the development view; class dependency reasoning is typically part of the Logical View; and asset management and branching strategies are explored using other views, which indicates multi-view correlation is required for build & release concerns; 3) typically, physical components and deployment nodes are rendered within a single integrated diagram, which indicates a spanning of the Development and Physical Views; 4) the Process View yields limited utility in most Enterprise IT contexts; more pragmatic representation is usually through the UC Realizations.

This paper proposes improvements to the 4+1 Views of Architecture based on the current state-of-the-practice. This amplification is being termed “The 3+1 Views of Architecture (in 3D)”. 
2. Introducing the “3+1 Views of Architecture”

The “3+1 Views of Architecture” is an architectural description viewpoint/view framework. It defines a suite of views and the typical diagrams that are created in practice within these views. It defines these views for communities of evaluating stakeholders to address their concerns related to a software solution. By interpreting 4+1 Views in 3D, one can better visualize the important relationships between the views. Taking the interpretation one step further, a 3 dimensional rendering also facilitates the notion of realization; that is moving from the abstract to the concrete. The “3” in the 3+1 Views represents three different structural viewpoints, specifically the Logical View, the Physical View and the Data View.

This amounts to a reduction of the “4” views to 2 generic views. Introduction of a separate and apart Data view yields the “3”. The “+1” denotes the same Use Case View, similar to the original Scenario View in 4+1. It is rendered in 3D to highlight the two dimensions of software intensive systems – dynamic/behavior and static/structure. Consistent with a Use Case-driven approach to arriving at a software solution, the Use Case View drives the construction of the structural views in a methodical way. The structural views represent the projections of the dynamics of the system of interest onto the structural “plane”, both logical and physical. The Data View represents a more concrete structural plane that implements persistence abstractions contained within the application tier. Figure 2 illustrates this interpretation of the 4+1 framework.

![Image](image.png)

**Figure 2. “3+1 Views of Architecture in 3D” – a pragmatic interpretation of “4+1 Views”**

Taking one iterative pass through the framework, Use Case types are identified within the Use Case View which represents observable value proposition requirement types that are identifiable by users of a
system. Deep-dives of these value propositions are taken through textual and/or visual representations as dynamic realizations, resulting in the identification of supporting abstractions. Typical UML diagrams leveraged for Use Case Realizations are Sequence Diagrams, Activity Diagrams and Communication Diagrams. From any of these models, object instances are identified and then projected onto structural entities or classes/types. Reasoning about the associations necessary among the classes/types assist with the identification of clusters of structural elements of the system. This then informs the architect about the coarse-grained logical abstractions and encapsulations of semantically related elements (subsystems) which are captured using Package Diagrams. This can be defined in a physical manner through the realization of packages into physical components and their component-oriented or service-oriented interfaces deployed on physical nodes within a physical communications topology.

3. Capturing levels of abstraction – "Altitude shifting"

Implicit within the dynamic dimension within the 3+1 Views articulation is the notion of “altitude”. By altitude we mean the various degrees of detail and work product formats/languages as one transitions from statements of need in the form of requirements artifacts to ground-level or where the rubber meets the road – code. Figure 3 illustrates the notion of shifting levels of detail in arriving at the concrete solution:

![Figure 3. Levels of abstraction within the 3+1 views framework](image)

One can see that the Use Case View represents one of the views that can be described at either the textual altitude, or at the technology neutral problem domain altitude, or more concretely at the technology specific solution domain altitude. Thus the Use Case view can be thought of as providing the bridge between the “what” and the “how”. Similarly, the Logical View typically has two differing altitude graduations as a result of the OOAD approach manifest within the RUP. These are known as the Analysis Model and the Design Model.

So through performing Use Case Analysis, one discovers the analysis or “what” classes that realize the Use Case Requirements. Once these abstractions have been discovered at the analysis level of detail, Use Case Design further elaborates the dynamics onto a structural plan in the form of design patterns and technology specific design or “how” classes.

4. Extension for Business-IT alignment Concerns and Enterprise architecture

One of the most powerful and yet overlooked aspects of the original 4+1 Views Framework is that it can apply to differing context-levels/scopes. The same technique of driving from the system use cases through their dynamics realizations onto a structural plane for systems architecture is the same technique applied for Business & Enterprise Architecture reasoning. The addition of 3 extra views in a similar 3-dimensional configuration results in a viewpoint framework that efficiently addresses the concerns of Business Architecture and Enterprise Architecture. Figure 4 illustrates this extension of applying 3+1 for purposes of Business Engineering and Enterprise Architecture.

![Figure 4. 3+1 Views applied to Business Architecture](image)

An object oriented approach [7] to business engineering and Business-IT alignment starts by describing significant design decisions related to
business value chains and the systems of systems that support the core value chains. The entry point view is the Business Use Case View. Here, the core value-add services are articulated to define the scope of a business from the perspective of the primary actor to a business – the customer. The realizations of the observable results of value for customers represent value chains. The dynamic interactions intersect with abstractions, whether business workers, business entities or business handlers in the Business Logical View. These three types are an application of Jacobson’s 3-types [8]. These abstractions are encapsulated in business packages, where logical cohesion exists thereby allowing the logical structure of the enterprise to be decomposed. This logical structure then is realized by physical structure in the form of business components deployed in physical geographical locations. This object oriented technique is not different than that for systems analysis & design, except the focus is on a larger context – that of business workers collaborating over time with systems of systems. Automated business workers represent systems in this approach, and messages encapsulate information flow among the objects of the business.

The UML is extended to integrate the object-oriented business engineering technique described above and integrated into the 3+1 Views through the stereotyping UML extensibility mechanism. This extension suite of notation and semantics is described in the UML Profile for Business Modeling [9]. The same notational semantics and idioms also can apply to the system-of systems and business architecture contexts. And more importantly, the correlation of views produced for system-level concerns are seamless with the views that describe encapsulations of elements described at the higher levels of abstraction.

5. Conclusion

This paper proposes enhancements to the 4+1 View Model of Architecture. The emphasis for value-add is from the perspective that any process should strive to make common sense and be rapidly applied. To this end, the original 4+1 Views approach to describing how to reason about and describe architecture has been interpreted in a new way. The “3+1 Views of Architecture (in 3D)” articulates the systematic way of building object-oriented and service-oriented systems intrinsic in the seminal 4+1 Views framework. This interpretation is intended to help a practitioner better understand and apply architectural description within common software delivery methods. The application of systems thinking to business engineering has also been integrated into this 3-dimensional viewpoint framework. The contributions come from pragmatic observations and experience from very large scale programs/projects. Future work related to the 3+1 viewpoint framework includes:

- Formally catalog 3+1 viewpoint configurations compliant with IEEE 1471 for single view concerns and consistent with UML 2.x;
- Define how to leverage perspectives [10] within the 3+1 framework to address multi-view correlation and cross-cutting concerns within 3+1;
- Compare and contract the 3+1 viewpoint framework against legacy Enterprise Architecture approaches including Zachmann [11], FEA [12], the ARIS House [13], and Rummler-Brache [14] – specifically around the integration of the process of developing the AD at both system and enterprise scope [15].

References

[15] Examples of enterprise-scope process descriptions include The Open Group Architecture Framework (TOGAF), and the Enterprise Unified Process (EUP) – EA Discipline.