Conceptual Framework of Indian Banking System in Recent Era: A Review

Abstract

With the expansion of network technology, banking technology has moved from PC-centric concept to the network-centric concept. Along with this, the way of architecting and designing software systems has changed. Each component of such systems is an independent subsystem which can be architected, designed, developed and managed separately.

This trend had a number of consequences for software engineering, in general, and software architecture, in particular. Architecting process for such systems became an essential part in software life cycle model. Architectural descriptions have long been recognized as an essential ingredient of a well-designed system. These descriptions mostly consist of a set of elements and interactions between them.

Banking Network System which provides web access for its users and uses services to communicate with other external devices, for example, ATMs. Online banking system which allows bank customers to have access to their bank accounts from any place in the world where they have internet connections and allows employees to serve clients using web access.

To sum up, Banking Network System:

- provides web access to bank customers
- provides web access to bank employees
- supports ATM access to the system

Keywords: Networks-Centric Architecture, Zachman Framework, Framework for Banking, Banking Network System.

Shamshir Ali
Research Scholar
Faculty,
Department of I.T.
M.D.D.M. College,
Muzaffarpur, Bihar

Introduction

Networks-Centric Architecture

Network-centric architectures describe software systems which are loosely-coupled, consist of independent components which more often is an integral of a larger system that involves the collaboration of software, hardware, and people to solve complex computing problems**1**.

The following sections describe major network-centric software-based system architectures.

Client-Server Architecture

Client-server architecture is a model of computing which splits computing tasks between client and server. Most often clients and servers are located on different hardware and are communicating over the network (Figure 1). In such type of architecture, clients request services and servers provide them to the clients.

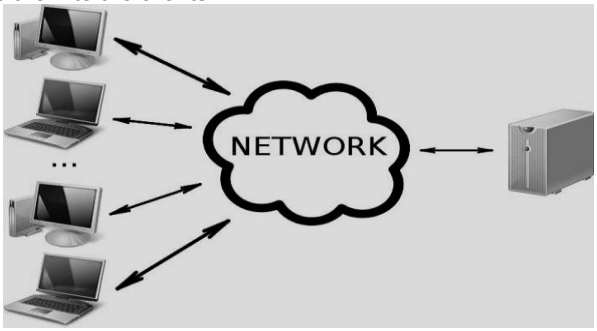


Figure 1. Client-server architecture

Two-tier client server architecture

Two-tier is the simplest type of client-server architecture. It basically defines how application processing is divided in the application. In two-tier application presentation layer is provided to multiple clients which communicate with the sever computer over the network. The server computer plays the role of centralized data storage (Figure 2).

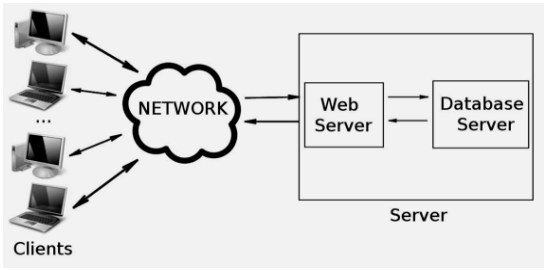


Figure 2. Two-tier client-server architecture

Most internet applications, such as email, telnet, ftp and a lot of web-sites are simple two-tier applications. Without providing a lot of data processing these applications provide a simple interface to access data over the network **2.**

Frameworks

Here we describe the most widely used and recognizable architecture frameworks. We also think that these architecture frameworks influenced the field of software-based system architecting the most. Therefore it is important to know and understand its concepts.

Zachman Framework

One of the first who raised a question about difficulty of managing information systems was John Zachman**3**. He identified the main contributing factors: the increasing size and the complexity of the systems, and the tendency of systems distribution because of enterprise operations automation.

Architecture Framework

Zachman originally explained his framework as an analogy with the building industry. He organized architectural artifacts in a two-dimensional schema. One dimension consists of:

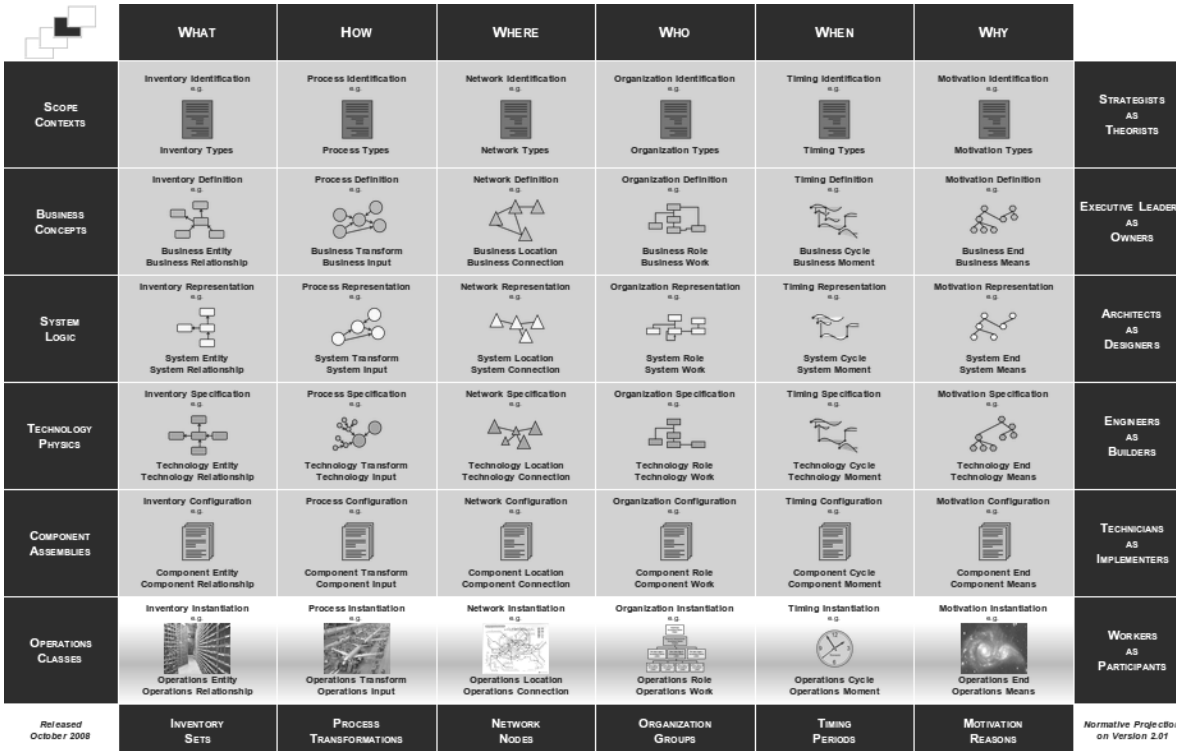
- Planner (Theorist)
- Owner
- Designer
- Builder
- Sub-contractor (Implementer)
- End User (Participant)

It represents a distinct view of the system during all phases of development. In other words every player requires complete information, but point of view of one player differs for another one. For example, “the owner is interested in a complete description of the functionality and aesthetics of the building. The builder is interested in a complete description of the materials and construction process. The owner doesn’t care about the placement of stubs in the walls. The builder doesn’t care how the bedroom windows line up with the morning sun”.From the IT prospective the owner is interested in a general outlook of what the product will look like. The designer is responsible for the logical view of the system. The builder is creating a physical layout of the system. Sub-contractor implements out-of-context sub components. And the end user has the final product.

The second dimension represents different descriptive focuses of each view. These six focuses are the next: what, how, where, who, when and why.

Each focus is different for each player, i.e. both the planner and the owner need to know *what*, but the planner’s “what” is different from owners “what”. And each of what, how, where, who, when and why depends on who is asking the question.

The current version (v2.01) is shown in Figure 10. It consists of 36 cells where the last row (6 cells) represents the final product, while other cells are for the product development.



"As we move horizontally in the grid, we see different descriptions of the system – all from the same player's perspective. As we move vertically in the grid, we see a single focus, but change the player from whose perspective we are viewing that focus".

Institute of Electrical and Electronics Engineers (IEEE) 1471 Standard

IEEE 1471 is an IEEE Standard for specifying the architecture of software-intensive systems. It is also known as ANSI/IEEE 1471-200, Recommended Practice for Architectural Description of Software-Intensive Systems. It was developed to provide a foundation for architecture specification of software-intensive systems. The standard is focused on "any system in which software development and/or integration are dominant considerations (i.e., most complex systems nowadays). This includes computer-based systems ranging from individual software applications, information systems, embedded systems, software product lines and product families and systems-of-systems"**4**.

Architectural Description

Each system has one or more stakeholders. And each stakeholder has his/her interests and concerns which are relative to the system. "Concerns are those interests which pertain to the system's development, its operation or any other aspects that are critical or otherwise important to one or more stakeholders.

The standard states that "every system has an architecture, in terms of its recommended practice" and that architecture can be recorded by an AD which is provided in Figure 4.

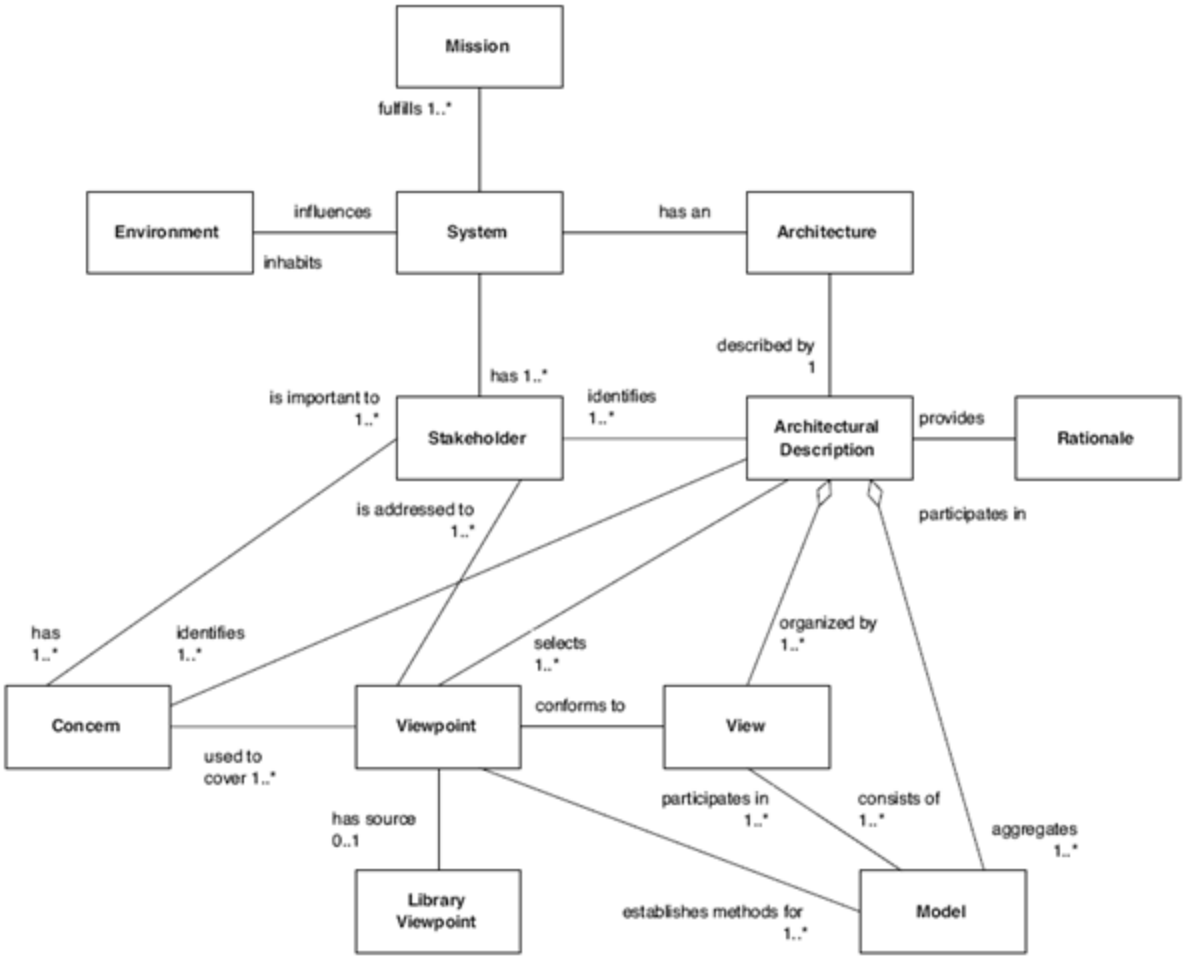


Figure 4. Conceptual Model of IEEE 1471 architectural description

View represents a system's architecture "with respect to a particular viewpoint". Where *viewpoint* – is "a pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis"

The Process of Creating System Architecture Specification and Simulation Model (Network Banking System in Recent Era)

Here we evaluate the applicability of the described Conceptual Framework and provide half a dozen examples of using Conceptual Framework for specifying network-centric system architectures. Also, for each example, we provide simulation models which were built based on the described system architecture specifications. Before we present examples of network-centric system architecture specifications, we need to describe the process of creating system architecture specification and its transformation to the simulation model.

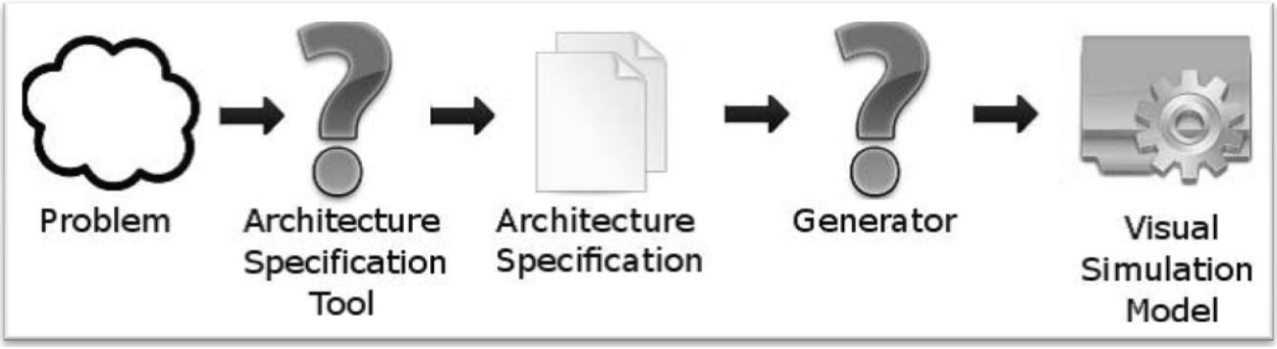


Figure 5. The process of creating system architecture specification and simulation model

At the beginning we have a problem. Then, using some architecture specification tool we create architecture specification. After that, some generator uses this specification to automatically build simulation model. But nowadays, there is no such generator which can get an architecture specification and build a simulation model from it.

In this chapter, to evaluate the applicability of the proposed Conceptual Framework, we have built architecture specification using Conceptual Framework manually. Moreover, we have built architecture specification using and JavaScript programming language. Then, for each example we manually created the simulation model, also using proposed Conceptual Framework. Simulation model is built in the Visual Simulation Environment (VSE)**.** The program which allows to build manually simulation models of different complexity and to specify programmatically its behavior.

Banking Network System

Banking Network System which provides web access for its users and uses services to communicate with other external devices, for example, ATMs. This example describes SOA of online banking system which allows bank customers to have access to their bank accounts from any place in the world where they have internet connections and allows employees to serve clients using web access.

To sum up, Banking Network System:

- provides web access to bank customers
- provides web access to bank employees
- supports ATM access to the system

System Architecture Specification

We can see four components there:

- Main Bank
- ATM
- Client
- Clerk

Client and Clerk communicate with Main Bank and each of them sends only requests. In the same time ATM communicates with Main Bank and sends requests and notifications. Each component has a description which contains name, description, services provided and services requested fields. And each of these fields provides corresponding information.

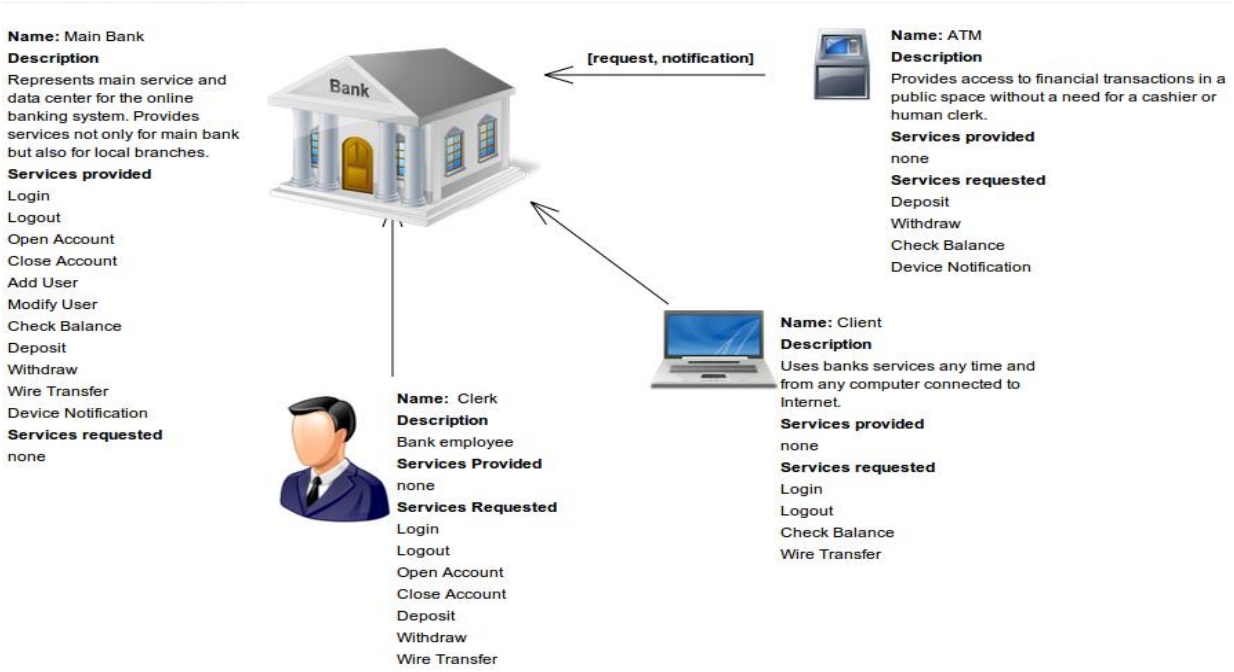


Figure 6. Top view of the Online Banking System architecture specification

"Application Interface" component sends only requests and "Status Component" component sends only notifications. We can also see that both components do not provide any services and request services of the Main Bank component.

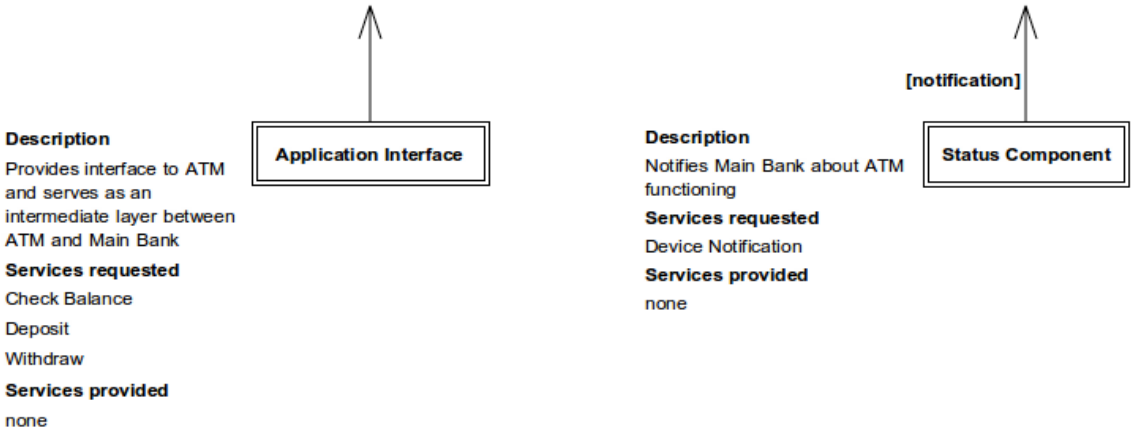


Figure 7. Specification of the ATM component

Reference

1. Chigani A. (2007), "Guiding Network-Centric Architectural Design: A Style-Based Approach".
2. Reese G(2000),Database programming with JDBC and Java, Second Edition, O's Reilley Media, Sebastopol, CA.
3. Zechman J. (1987),"A Framework for information systems architecture", IBM Systems Journal 26,3, pp. 276-292.
4. IEEE (2000),"ANSI/IEEE Std 1471-2000 Recommended Practice for Architectural Description of Software-Intensive Systems", IEEE Architecture Working Group, October 2000.