

# A Survey on Distributed System Applications Using Design Pattern

<sup>1</sup>Jayashree D. Jadhav, <sup>2</sup>Manjusha Joshi, <sup>3</sup>Dr.S.D.Joshi, <sup>4</sup>Dr. R.M.Jalnekar

<sup>1,2,3</sup>BVDUCOE, Pune, Maharashtra, India

<sup>4</sup>VIT, Pune, Maharashtra, India

## Abstract

Distributed systems envelop numerous ranges of computer science, for example machine architecture, networking, working frameworks, embedded devices, and security. On top of new technologies, such as remote and wearable Pcs, conveyed frameworks saturate regular life—in homes and at work and at live up to expectations. Different parts of distributed computing, for example performance, security, and unwavering quality, can effectively verify the victory or failure of numerous organizations, urban areas, or even nations. In spite of the fact that there are a hefty number of software development systems for standalone software, small endeavour is being paid into examining specialised techniques that focus on the advancement of Distributed Applications (DAS) in the time of Internet and web-based provisions. The key idea of this paper to study different latest applications which are developed for Distributed system using design patterns. This paper is also specifying working, advantages and weaknesses of some latest distributed system applications developed using different design patterns.

## Keywords

DAS, DS

## I. Introduction

A decade prior, most software comprised of standalone applications and just negligible types of cooperation's through documents and information turned exchange methodologies were underpinned. With the pervasive utilization of the web and its cohorted technologies, more and more requisitions that are "participating" over the Web tend to be created. Relying on the structure and reason for the aforementioned requisitions, they exist under diverse sections for example integrated applications-business/B2B requisitions, Web requisitions, and so on [1]. Distributed Applications are now days used in various domains. Some of such popular domains are given below:

### A. Web Services

The web and intranets are the best examples of distributed systems. Today Internet and different Web services are popularly used by number of users. Internet is making use of different platforms, protocols and different design patterns and security mechanisms. Internet can be considered as the one of the most popular Distributed System used today and is growing day by day using some new technologies, new patterns..etc.

### B. Banking

Another popular Domain making use of Distributed System Applications is Banking. Different banks provide various services like registration, Checking Account details...etc for its different users. As users are located anywhere in the world, the system running Banking Application should Support Distributed Capabilities. Today Various Banking Applications are available and still developing with the use of different Platforms, different languages, different security mechanisms and design patterns.

### C. Enterprise Applications

Enterprise Applications that is Applications that are run on the local network of Enterprises that is Intranets. Generally in Large Organizations various Departments are making use of different software that may be built on different platforms, may have different design patterns...etc. So Such Enterprises and need a Distributed Application which will handle such diversity in using Enterprises software.

Lot of Such like Distributed applications like ERP, CRM are available which are being used in Enterprises. Various improvements in performance of this software can be done by using some new design patterns.

### D. Medical or Healthcare

Distributed System Applications are also popular in Medical field. These Applications are used for various purposes like Sharing digital information and computer resources among distant clinical and research facilities, for medical image transfer, telemedicine or Electronic Health Care..etc. [7].

## II. Background

What is distributed system?

Distributed systems are groups of networked computers which have the same objective for their function. This kind of system has the following characteristics:-

- Comprises of loosely coupled sites that share no physical component
- Run on every site are independent of each other
- Transactions might access information at one or more sites

Distributed system is classified into 2 types. Homogeneous & heterogeneous distributed system.

In homogeneous DS, all sites they have identical software & it appears to user as single system. All sites are known to each other so that they agree to cooperate with user request.

In heterogeneous DS, different sites use different schemas & software and may not be aware of each other & they provide only limited facilities for cooperation in transaction processing.

### A. Architecture of Distributed System

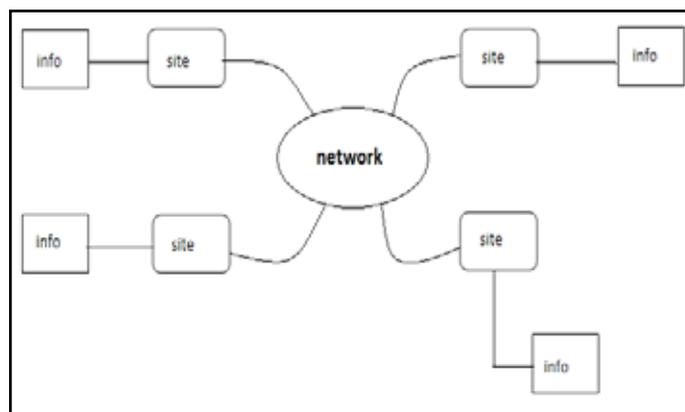


Fig. 1: Architecture of Distributed System

## B. Components of Distributed System

### 1. Computer Workstations (Sites or Nodes)

A distributed DBMS consists of a number of computer workstations that form the network system. The distributed database system must be independent of the computer system hardware.

### 2. Network Components (Both Hardware and Software)

Each workstation in a distributed system contains a number of network hardware and software components. These components allow each site to interact and exchange data with each other site. Network system independence is a desirable property of the distributed system.

### 3. Communication Media

In a distributed system, any type of communication (data transfer, information exchange) among nodes is carried out through communication media. This is a very important component of a distributed DBMS. It is desirable that a distributed DBMS be communication media independent, that is, it must be able to support several types of communication media.

### 4. Transaction Processor (TP)

A TP is a software component that resides in each computer connected with the distributed system and is responsible for receiving and processing both local and remote applications' data requests. This component is also known as the application processor (AP) or The transaction Manager (TM).

### 5. Data Processor (DP)

A DP is also a software component that resides in each computer connected with the distributed system and stores and retrieves data located at that site. The DP is also known as the data manager (DM). In a distributed DBMS, a DP may be a centralized DBMS.

## C. Reason for Distributed System

Simply on the grounds that it is simple and reasonable to associate numerous machines together does not vitally imply that it is a great thought to do so. There are genuine benefits in building distributed systems.

### 1. Distributing Machines May Make Sense

It makes sense to put the CPUs for ATM cash machines at the source, every coordinates with the bank. Every bank can have one or more Pcs coordinates with one another and with different banks. For computer graphics, it makes well to put the design preparing at the client's terminal to maximize the bandwidth between the mechanism and processor.

### 2. Cooperative and Social Networking

Clients that are geographically differentiated can now work and play together. Examples of this are plenty: distributed document systems, audio/video conferencing, message, multiplayer games, barbers, and social networks.

### 3. Increased Reliability

Assuming that a small percentage of machines break, whatever remains of the framework remains whole and can do useful work.

## 4. Incremental Growth

A company may purchase a machine. Eventually the workload is too great for the machine. The main alternative is to replace the Pc with a faster one. Networking allows permits you to add onto an existing infrastructure.

## 5. Remote Services

Users may need to access information held by others at their systems. Examples of this include web browsing, remote file access, and programs to retrieve large files.

## 6. Mobility

Clients move around with their Pcs and cell phones. It is not feasible for them to carry all the qualified information they require with them.

By using various design patterns we can build distributed system.

## D. Design Pattern

Design patterns are recurring solutions to software design problems you find again and again in real-world application development. Patterns are about design and interaction of objects, as well as providing a communication platform concerning elegant, reusable solutions to commonly encountered programming challenges [8].

### 1. Business Process Modelling Pattern

Business Process Modelling is turning into a higher necessity for business administrators and analysts as there is an expanding attention in organizations to document, understand and enhance their business processes. Generally recorded and streamlined processes are vital in a smooth working of a team as they help ensure consistency, track ability and focus towards a shared goal. Business Process Modelling is a combo of different process related steps for example Process Mapping, Process Discovery, Process Simulation, Process Analysis and Process Improvement [5].

### 2. Service Oriented Architectural Pattern

SOA is the aggregation of components that fulfil a business need. It involves components, services, and processes. Components are binaries that have a described interface and a service is a grouping of components to finish the employment.

A service is generally implemented as a coarse-grained, discoverable software entity that exists as a single instance and interacts with applications and other services through a loosely coupled (often asynchronous), message-based communication model.

The most essential part of SOA is that it differentiates the service's implementation from its interface. Service customers view a service essentially as a communication endpoint supporting a specific request format or contract. How service executes service requested by buyers is irrelevant; the main mandatory necessity is that the service sends the response back to the buyer in the agreed format, specified in contract [6].

### 3. Technology Based Pattern

Pattern-based strategy gives a schema to proactively look for examples from traditional and non-traditional information sources, model their effect on the organization, and after that quickly adjusts methodology and business activities dependent upon the effect of these developing examples.

Pattern-based strategy does not replace existing strategy management methodologies, such as the balanced scorecard or value-based management, but instead complements them with a strong focus on the enabling role of technology in strategy management.

### III. Related Work

Software design patterns are best practice solutions to common software problems. The thought of building software architectures from software design patterns is not new. There are numerous eminent approaches for software architectures from design patterns and are in particular real time systems these approaches just give unique descriptions of design patterns and high level guidance on how the patterns can be used to form software architectures [3]. The key idea of this paper is to present the ways of enhancing quality of design procedure & expanding reusability of industrial control system components by means of applying design patterns for development of system software architecture [2].

To build distributed system & web applications the middleware methods & design methods are used. The middleware methods consist of communication, content & business process levels. At communication level, there are technologies that support tightly or loosely coupled communication styles. Electronic data interchange & internet based services are supported by communication & content levels. The workflow based system focus on interoperability at business process level. The various design methods are specifically proposed for concurrent & distributed systems. Due to their support for modularity, flexibility & reusability the object oriented methodologies have been proposed for the design of DAs [1].

### IV. Issues & Challenges

Different kinds of distributed systems work today, every pointed at tackling various types of issues. The challenges faced in building a distributed system vary depending on the requirements of the system. In general, on the other hand, most systems will need to handle the following issues [9-10].

#### A. Heterogeneity

Different substances in the system must have the capacity to interoperate with each one in turn, despite distinctions in hardware architectures, operating systems, communication protocols, programming languages, software interfaces, security models, and data formats.

#### B. Transparency

The whole framework should appear as a single unit and the complexity and interactions between the components should be typically hidden from the end user.

#### C. Fault Tolerance and Failure Management

Failure of one or more components should not bring down the whole framework or entire system, and should be isolated.

#### D. Scalability

The system may as well work effectively with expanding number of clients and expansion of a resource may as well enhance the performance of the system.

#### E. Concurrency

Shared access to resources should be made possible.

#### F. Openness and Extensibility

Interfaces should be cleanly differentiated and openly accessible to empower simple extensions to existing components and include new components.

#### G. Migration and Load Balancing

Permit the development of undertakings inside a framework without influencing the operation of clients or applications, and distribute load among available resources for enhancing performance.

#### H. Security

Access to resources should be secured to guarantee just known clients have the ability to perform permitted operations.

#### V. Conclusion

As the popularity of distributed system applications increasing day by day. Developers are focusing on the advancements of distributed system applications using various techniques. Design patterns are one of such technology that is used in the development of such distributed applications. In this paper we have presented some information about distributed system applications like the domains in which it is used, the advantages of using it, the problems in designing the distributed system applications...etc

#### References

- [1] F.Dabous, F.Rabhi, H.Yu, "Using Software Architectures and Design Patterns For Developing Distributed applications", IEEE-(ASWEC-2004)
- [2] E. Burmakin, B. Krassi, "Design Patterns for development of Dynamic Distributed Automation System", Helsinki university of technology.
- [3] J.S.Fant, "Building Domain specific Software architecture from Software architectural design Patterns", ICSE-2011
- [4] C. Tianhuang, H. Feifei, "Design Patterns Application in a distributed system", IEEE-2010.
- [5] Thomas Gschwind, Jana Koehler, Janette Wong, "Applying Patterns during Business Process Modeling".
- [6] Duane Nickul, Laurel Reitman, James Ward, Jack Wilber, "Service Oriented Architecture (SOA) and Specialized Messaging Patterns", 2007.
- [7] Hatem M. Hamad, "Distributed electronic health care system (DEHCA)", Islamic university, 2003.
- [8] Ladan Tahvildari, "Accessing the impact of using design pattern based system", 1999.
- [9] Petru Eles, "Distributed system".
- [10] Krishna Nadiminti, Marcos Dias de Assunção, Rajkumar Buyya, "Distributed Systems and Recent Innovations: Challenges and Benefits".
- [11] Andy Crabtree, Terry Hemmings, Tom Rodden, "Pattern-based Support for Interactive Design in Domestic Settings", ACM-2002.
- [12] FRED B. Schneider, "On Distributed Computing", IEEE DS Online, Vol. 1, No. 1, Interview by Dejan Milojevic, [Online] Available: [dejan@spica.hpl.hp.com](mailto:dejan@spica.hpl.hp.com)



Mrs. Jayashree D. Jadhav received her B.E. degree in Computer F Rajarambapu institute of technology, Sakharale in 2008. She is currently, working as the Research Student in Computer Engineering Department Bharati Vidyapeeth Deemed University College of Engineering, Pune. Her research interests include software engineering and Distributed Systems.



Dr. Shashank Joshi received his B.E. degree in Electronics and Telecommunication from Govt. College of Engineering, Pune in 1988, the M.E. and Ph.D. Degree in Computer Engineering from Bharati Vidyapeeth Deemed University Pune. He is currently working as the Professor in Computer Engineering Department Bharati Vidyapeeth Deemed University College of Engineering, Pune. His research interests include software engineering.



Ms. Manjusha Joshi received her B.E. degree in Computer from Walchand College of Engineering, Sangli in 1991, the M.E. and Ph.D Degree in Computer Engineering from Bharati Vidyapeeth Deemed University Pune. She is currently, working as the Research Student in Computer Engineering Department Bharati Vidyapeeth Deemed University College of Engineering, Pune. Her research interests include software engineering and Distributed Systems.



Dr. Rajesh Jalnekar received B.E. degree in Electronics and Telecommunication from Govt. College of Engineering, Pune in 1988, the M.E. and Ph.D. Degree in E & TC from Pune University, Pune. He is currently working as the Director, Vishwakarma Institute of Technology, Pune. His research interests include Distributed Systems.