TOWSON UNIVERSITY

COLLEGE OF GRADUATE EDUCATION AND RESEARCH

INVESTIGATION & COLLECTION OF OPERATING SYSTEM CONSTRUCTS NEEDED FOR MIDDLEWARE SOLUTIONS

BY

REGULAPATI VENKATA RAMANA RAO

A PROJECT SUBMITTED TO THE FACULTY OF

TOWSON UNIVERISTY IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN COMPUTER SCIENCE

AUGUST 2005

TOWSON UNIVERSITY

TOWSN, MARYLAND 21252

INVESTIGATION & COLLECTION OF OPERATING SYSTEM CONSTRUCTS NEEDED FOR MIDDLEWARE SOLUTIONS

Abstract
Development of large class of distributed systems can be simplified by leveraging middleware, which is layered between network operating systems and application components. In order for the distributed component system to appear as an integrated computing environment, the components have to communicate with each other. This kind of communication can only be achieved by using network protocols, which are often classified by the ISO/OSI reference model. These distributed systems usually built on the top of the transport layer, of which TCP or UDP are good examples. The layers underneath are provided by the network operating system. Different transport protocols have in common that they can transmit messages between different hosts. This paper provides a through investigation & collection of different operating system constructs those needed for various middleware solutions.

1. INTRODUCTION

Even though the first wave of client/server applications has been successful in many aspects, often these applications did not face the challenge of scalability. As companies are tried to grow these applications from departmental usage to enterprise usage, the challenges of scalability are just around the corner. Many industry experts agreed that middleware is the answer for the application integration and scalability problems faced by the most of the companies. In the beginning, middleware evolved around the database access model. This database model is usually characterized by two tier applications, in which a fat client application retrieved and updated information that was maintained by a relational database. Early solutions were proprietary in nature. Then the SQL standard emerged and provided a common language for data access from different databases. Finally ODBC set a de-facto standard for database access.

Organizations that have a strong need for scalability have begun to build applications following a new model; that of distributed application components mostly in the form of a three-tier architecture. This architecture extends the two-tier model by adding another tier in the middle between client and server. This tier is called an application server. In this model client and server become thinner and the business logic moved from client to middle tier. As the move to three tier model continues, middleware is evolving to support this new model as well. Early middleware products capable of supporting the three tier model did not provide mush more than a higher level abstraction of communication protocols. Using this approach the application developer could use a higher level API instead of a low level API like TCP/IP sockets

The distributed computing environment form Open group was the first vendor technology which provided not just a common set of APIs and a consistent programming model across multiple operating platforms, but also a comprehensive set of distributed computing services.

There were several middleware solutions available in the market serving different purpose. Some of them are CORBA, RMI, DCOM ,COM, OLE, MQSeries, .NET, JMS, J2EE, ADO, ODBC, JDBC etc. This paper mainly focused on investigating and collecting different operating systems constructs needed for various middleware solutions. In this process I did lot of literature search and gathered hundreds of middleware papers pertaining to different categories and after reading all of them I have selected seventy papers which were found good for my project. In my investigation, I tried to collect the following artifacts from these selected papers: 1) Summary, 2) Technologies used, 3) Middleware Category and 4) OS Constructs. The investigation report on all these papers is followed:

2. INVESTIGATION

Paper# 1:

A Middleware system which Intelligently Query Results

Summary:

Describes how caching was used to improve performance in the Accessible Business Rules (ABR) framework for IBM's Websphere. ABR is a middleware system, which enables application writers to build applications where the time and situation-variable parts of their business logic are extremely applied entities known as business rules.

Middleware

Database Middleware

Technologies

Uses Java, JSP/Servlet, JDBC, RMI

OS Constructs

RMI/IIOP, Sockets

Paper# 2:

MOCHA: A Self-Extensible Database Middleware System for Distributed Data Sources

Summary:

Is an alternative database middleware solutions to the problem of integrating data sources distributed over a network In existing Middleware solutions where user defined functionality for data types and query operators is manually deployed, is inadequate and will not scale to environments with hundreds of data sources. The high cost and complexity involved in having administrators installing and maintaining the necessary software libraries into every site in the system makes such approach impractical. MOCHA is a self-extensible middleware system written in JAVA in which user defined functionality is automatically by the system to the sites that do not provide it. This is realized by shipping Java classes implementing the required functionality to the remote sites.

Middleware

Database middleware

Technologies

JAVA, XML, JNI, JDBC, DOM, RMI

OS Constructs

RMI/Sockets

Paper# 3

Adaptable Query Optimization and Evaluation in Temporal Middleware

Summary

This paper offers a temporal middleware approach to building temporal query language support on top of conventional DBMSs. Unlike other approaches this middleware performs some query optimization, thus dividing the query processing between itself and the DBMS, and then coordinates and takes part in the query evaluation. The architecture proposed is called TANGO (Temporal Adaptive Next-Generation query Optimizer and processor)

Middleware

Database Middleware

Technologies

C/C++, JAVA, XXL, JDBC

OS Constructs

RPC

Paper# 4

A Middleware Implementation of Active Rules for ODBMS

Summary

A middleware approach for supporting active rules is presented in this paper. A prototype of this system has been implemented as a component of a larger middleware system that provides ODMG interfaces and Spatial extensions to ODBMS. The active rules are ECA type and used for integrity constraints in this system. By implementing as Middleware, heterogeneous ODBMS in distributed environment can be provided an active rule system in system independent manner.

Middleware
Database middleware

Technologies
CORBA, OMG IDL,

OS Constructs
Sockets if CORBA implemented in JAVA

Paper# 5

A Configurable Multimedia Middleware platform (GOPI)

Summary
The **GOPI** (Generic Object Platform Infrastructure) middleware platform supports multimedia applications in a standard operating system environment. This is mainly focused on dynamic QoS management issues. In particular, it describes low level concurrency and communications mechanisms designed to maximize performance and predictability in multimedia middleware. GOPI attempts to deliver performance and predictability in a configurable manner in a standard operating system environment. This is also designed with backward compatibility to CORBA

Middleware
Multimedia Middleware

Technologies
C, Runs on UNIX

OS Constructs
Sockets

Paper# 6

A multi tier framework for accessing distributed, heterogeneous spatial data in a federation based EIS

Summary
The goal of the framework is to use new technologies and concepts like JAVA/RMI, CORBA, to develop generic Geographical Information System (GIS) components that can be easily reached from end users via WWW, easily integrated into component based applications and internet based information system architectures. The framework is intended to provide a clear separation between application development and data sources.

Middleware
Distributed Object Middleware

Technologies
JAVA/RMI, SQL*Net etc

OS Construct
Sockets

Paper# 7

Exploiting Reflection in Mobile Computing Middleware (CARISMA)

Summary
CARISMA is a project carried out at University of London, is a middleware that uses both structural and behavioral reflection to enable context-aware in between mobile applications. The middleware is in charge of maintaining a valid representation of the execution context, by directly interacting with the underlying network operating system.

Middleware
Reflective Middleware

Technologies
JAVA, XML

OS Constructs
Sockets

Paper# 8

A CORBA Service Enabling Programmer-Friendly Object-Oriented Real-Time Distributed Computing (TMOES)

Summary
This paper discusses a cost-effective approach for facilitating CORBA-compliant TMO (Time-triggered Message-triggered Object) Structured Real Time distributed application programming. It is realized via provision of middleware and does not require any change in the ORB core. The initiation and scheduling of the executions of TMO methods are managed by TMOES (Time-triggered Message-triggered Object Execution Support) AnyORB which is a CORBA compliant derivation from the TMO support middleware model.

Middleware
Distributed Object Middleware

Technologies

C++, CORBA

OS Constructs

RPC

Paper# 9

Integration of QoS-Enabled Distributed Object Computing Middleware for Developing Next-Generation Distributed Application

Summary

This paper describes the integration of QoS-enabled distributed object computing (DOC) middleware for developing next-generation distributed applications. QoS-enabled DOC middleware facilitates ease of development and deployment of applications that can leverage the underlying technology or end-stream QoS architecture. QuO is a distributed object computing (DOC) framework designed to develop distributed applications that can specify the QoS requirements, the system elements that must be monitored and controlled to measure and provide QoS and the behavior for adopting to QoS variations that occur at run-time.

Middleware

Distributed Object middleware

Technologies

CORBA

OS Constructs

RPC

Paper# 10

A Combat Management System Middleware based on CORBA

Summary

Turkish navy has started an initiative to build a new middleware as part of a generic CMS development project named GENESIS. This middleware satisfies the requirements of CMS applications that run in a heterogeneous distributed computing environment (DCE). The approach was to use off-the-shelf components and open standards to build a portable and maintainable middleware while preserving functionality, performance and reliability requirements of CMS applications. The focus of this paper is the framework, GENESIS Information Exchange (GENIE), which is based on open standards and OTS. GENIE provides runtime support for CMS applications which require high performance and reliability. It should also maintain a good degree of flexibility for existing applications. GENIE is a LAN oriented system that performs its functionality on a number of distributed nodes. GENIE has three layers 1) Communications and Database library interface (CANDLE), 2) Database and connector and 3) OS and Network abstraction layer.

Middleware

Distributed Object Middleware

Technologies

CORBA, C++, CORBA Naming Service, Sun Solaris platform

OS Constructs

RPC

Paper# 11

An Object Infrastructure For Internet Middleware

Summary

IBM's component broker is Internet middleware for distributed objects. It is designed to absorb complexity in the operational and development environment so that applications appear to be transparent, local and fully resourced. Component broker is a complete package with everything a developer needs to build, run, and manage web-based business objects, components and applications. It consists of tools to build distributed objects, business objects and applications. Component broker is a logical object server that pulls the various capabilities of an object infrastructure into a cohesive environment. When an application is designed and developed using the component broker tools, it fits into the environment and can use all its capabilities.

Middleware

Internet Middleware

Technologies

CORBA, ORB

OS Constructs

RPC/Sockets

Paper# 12

Applying a Scalable CORBA Event Service to large-Scale Distributed Interactive Simulations

Summary

The CORBA event service provides a flexible model for asynchronous communication among distributed and collocated objects. This paper makes five contributions to the design, implementation and performance measurement of distributed interactive simulation systems. Many distributed applications exchange asynchronous requests usig event-based execution models. To support these common use-cases the OMG defined a CORBA Event service component in the CORBA Object Services (COS) layer. The standard COS event service specification lacks several important features required by large-scale distributed interactive simulations. Chief among these missing features are centralized event filtering, efficient and predictable event dispatching, efficient use of network and computational resources. To resolve these limitations this paper addresses a Real time Event service as part of the TAO project. TAO's Real time event service extends the COS Event service specification to satisfy the quality of service needs of real-time applications in many domains.

Middleware

Distributed Object Middleware

Technologies

CORBA, C++

OS Constructs

RPC

Paper# 13

A CORBA-Based Application Service Middleware Architecture and Implementation (CASM)

Summary

This paper is focused on a distributed architecture that respects the logical constraints of different domains, allowing exchange of management information and decision making as to the possibility of remote access that can satisfy the application requirements of QoS. This framework supports management operations to guarantee QoS for a variety of resource types, while processing a uniform operational interface at the application layer. The architecture depends on distributed computing principles; however it supports hierarchical support for the administration of different domains, which can result in load balancing and effective resource usage even for extended LANs when the user properties allow it. This design is governed by three goals: 1) To serve as a better than the best effort middleware to the application layer, 2) To reflect the application requirement with respect to the usage control of resources, 3) To require only minor changes to existing applications in order to achieve benefits from CASM.

Middleware

Distributed Object Middleware

Technologies

CORBA, C++

OS Constructs

RPC

Paper# 14

Applying Reflective Middleware Techniques to Optimize a QoS-enabled CORBA Component Model Implementation

Summary

Real time CORBA ad CORBA messaging, address many end-to-end QoS Properties, they do not define strategies for configuring these properties into application flexibly, transparently and adaptively. Therefore application developers must take these configuration decisions manually and explicitly, which is tedious, error-prone, and often sub-optimal. This paper presents three contributions to the study of middleware for QoS-enabled component based applications. It outlines reflective middleware techniques designed to adaptively 1) Select optimal communication mechanisms 2) Mange QoS properties of CORBA components in their containers and 3) Configure selected components executors dynamically.

Middleware

Reflective Qos-Enabled Middleware

Technologies

CORBA, C++

OS Constructs

RPC

Paper# 15

The Role of Software Architecture in Constraining Adaptation in Component-based Middleware platforms

Summary

This paper discusses the role of software architecture in maintaining the overall integrity of the system in such an environment. More specifically, the paper discusses extensions to the Aster framework to support the re-configuration of a reflective middleware platform in a constrained manner.

Middleware

Reflective middleware

Technologies

CORBA, C++

OS Constructs

RPC

Paper# 16

Systematic Aid for Developing Middleware Architectures

Summary

Most of the existing middleware architectures provide highly flexible and reusable functionality that can be composed in various ways toward satisfying certain nonfunctional requirements. But mastering and exploiting this flexibility is time consuming and problematic, even for middleware experts. This paper developed an environment that facilitates the design and quality analysis of flexible and configurable middleware architectures, providing three main features. 1) Architectural description language: helps to model middleware architectures, 2) Repository of architectural description: Repository is populated with architectural descriptions of middleware infrastructures that can include the specification of the basic services provided by the infrastructure, as well as related constraints, 3) Automated tool support: helps to construct all possible valid compositions of a given set of interaction patterns, each describing how to use the elements of a middleware service in the interests of providing a particular nonfunctional property.

Middleware

Configuration Middleware

Technologies

ADL, CORBA etc

OS Constructs
RPC/Sockets

Paper# 17

Frameworks for Component-Based Client/Server Computing

Summary
This paper is introducing the basics of client/server computing and component technologies and then proposed two frameworks for client/server computing using distributed objects. The author discussed topics affecting client/server frameworks, with a focus on the delegation of responsibilities between clients and servers and the stratification of client/server systems into levels. The author discussed CORBA from OMG and Microsoft's DCOM.

Middleware
Distributed Object Middleware

Technologies
CORBA, Java, J++, DCOM, COM, VC++, C++

OS Constructs
RPC

Paper# 18

A Reflective Component-Based & Architecture Aware Framework to Manage Architecture Composition

Summary
This paper proposed a reflective component-based framework with architecture style awareness for managing architecture composition and constraining adaptation. Specifically this framework offers a principled way to deal with both introspection and adaptation of basic and composite components. It provides the developers with the ability to choose, extend and modify architecture style managers. These managers are responsible to represent and check architecture constraints both at development and deployment time

Middleware
Configuration Middleware

Technologies
Java, RMI

OS Constructs

Sockets

Paper# 19

Patterns, Frameworks, and Middleware: Their Synergistic Relationships

Summary

Patterns, frameworks and middleware are increasingly popular techniques for addressing key aspects of the challenges for software development. Patterns codify reusable design expertise that provides time-proven solutions to commonly occurring software problems that arise in particular contexts and domains. Frameworks provide both a reusable, product architecture – guided patterns – for a family of related applications and an integrated set of collaborating components that implement concrete realizations of the architecture. Middleware reusable software that leverages patterns and frameworks to bridge the gap between the functional requirements of applications and underlying Operating systems, network protocols and databases. This paper presents an overview of these things and how these technologies complement each other to enhance reuse and productivity. To explain these concepts this paper focused on The ACE ORB (TAO) middleware as a case study to illustrate how patterns and frameworks can help middleware developers build and evolve software by reducing the coupling between its components. TAO is a high-performance, real-time implementation of the CORBA 3.0 specification that supports the distribution middleware capabilities.

Middleware

Distributed Object Middleware

Technologies

CORBA, C++

OS Constructs

RPC

Paper# 20

Managing Complexity: Middleware Explained

Summary

This paper broadly discussed about the distributed platforms CORBA, RMI and DCOM those become quite popular. These platforms extend earlier distributed component technologies like DCE, RPC and NetWare that weren't object based. The characteristics of these platforms includes: 1) masking heterogeneity in the underlying infrastructure by cloaking system specifics, 2) permitting heterogeneity at the application level by allowing the various components of the distributed application to be written in any suitable language, 3) providing structures for distributed components by adopting object-oriented principles, 4) offering invisible, behind-the-scenes distribution as well as the ability to know what's happening behind the scenes; and 5) providing general-purpose distributed services that aid application development and deployment.

Middleware

Distributed Object Middleware

Technologies

CORBA, DCOM, RMI

OS Constructs

RPC/Sockets

Paper# 21

The Design of an Application Programming Interface for QoS-Enabled Multimedia Middleware

Summary

This paper discussed about the development and implementation of an API which supports flexible communication services handling QoS specifications, transparently hiding easy-to-easy upper API from a programmer's perspective achieves the required degree of simplicity. A highly flexible configuration file passes application QoS requirements via the API from applications to the communication subsystem. Efficient process to process data transfers are supported by the data delivery protocol. By bypassing isochronous data to devices directly the API satisfies multimedia applications causing significant improvements of data management efficiency during data transfer phase. This interface for advanced multimedia middleware hides away communication relevant information from applications and provides a minimal set of interface functions and operations.

Middleware

Multimedia Middleware

Technologies

C, C++

OS Constructs

BSD Sockets

Paper# 22

The Architecture Of Global Access

Summary

This paper discus the parts of Internet and its components like, User and real world interfaces, Applications, Data, Delivery System and Middleware. If the internet were a designated artifact – like the Golden Gate Bridge – then these components would indeed suffice. When we consider above components, first note how the major corporate players have followed the standard chess strategy of controlling the center – in this case, Middleware. Microsoft, from its dominant position in delivery systems has been developing its DCOM etc

Middleware

Distributed Object Middleware

Technologies

DCOM

OS Constructs

RPC

Paper# 23

OFTT: A Fault Tolerance Middleware Toolkit for Process Monitoring And Control Windows NT Applications

Summary

This paper presents OFTT (OLE Fault Tolerance Technology), a fault tolerance middleware toolkit running on Windows NT Operating system that provides required fault tolerance for networked PCs in the context of industrial process monitoring and control applications. It is based on COM technology.

Middleware

Distributed Object Middleware

Technologies

COM, OLE

OS Constructs

RPC

Paper# 24

Global Change Master Directory: Object Oriented Active Asynchronous Transaction Management in a Federated Environment Using Data Agents

Summary

The Global Change Master Directory (GCMD) is an earth science information repository that tracks research data on global climatic change. Building a directory of earth science metadata that allows the exchange of metadata content among partner organizations is challenging because of the complex issues involved in supporting heterogeneous metadata schema, database schema, database implementation and platforms. This paper describes the design of MD8 (master directory v8.0) which allows automated exchange of metadata content among earth science collaborators by implementing an asynchronous distribution transaction protocol. The system includes a Local Data Agent for each partner that captures database updates and broadcasts them to other nodes through an announcer program asynchronously.

Middleware

Agent Based Middleware

Technologies

Java, RMI, JDBC

OS Constructs

Sockets

Paper# 25

An Agent Framework Based on Distributed Object

Summary

Currently there are many researches that focus on using agent and multi-agent system theories to build distributed applications. Constructing applications especially distributed applications by agent abilities can get strongpoint. In this paper the author discussed some related concepts of distributed systems based on agent and multi-agent systems are proposed. He discussed an agent communication language called KQML (Knowledge Query and Manipulation Language). Through KQML, agents could conveniently share and interchange knowledge each other. Implementation of Agent Communication Object(ACO) for KQML is based on ORB/IIOP.

Middleware

Agent Based Middleware

Technologies
KQML

OS Constructs
ORB/IIOP

Paper# 26

Developing Middleware for Web-aware Systems: Lessons Learned

Summary
This paper discussed the design and development issues concerning middleware for web-based systems. It presented a new system called GoWeb as a distributed infrastructure where web-enabled applications can be embedded. GoWeb supports collaborative work with multimedia user interface. It is object-oriented distributed software that gathers objects into a web-enabled application and can be accessed from any platform that supports JVM.

Middleware
Web-based Middleware

Technologies
Java, CORBA, RMI, JavaScript

OS Constructs
Sockets

Paper# 27

Java Servlets and Enterprise Java Beans in Enterprise Architectures: Friends or Foes Part II

Summary
In this article the authors tried to tackle a question that we have been asked repeatedly during the past few years. As J2EE-based application servers like BEA's Weblogic server and IBM's Websphere gain more widespread acceptance, should Java servlets or EJBs be used as a foundation for e-business applications? Most of the early web-based applications have been developed in servlets. First the servlet model offers more simplicity than EJBs and is easier foe IT developers and mangers to comprehend as they get stated with technology. Mean while the vendors stated that the purpose of J2EE application servers is to support high-end enterprise applications and to serve as the new strategic middleware platform for all the application development and deployment. It is there fore important to understand the difference in capabilities of servlets vs EJBs so that the suitability of one approach over the other can be determined given the requirements of a particular project, the readiness of IT personnel, budgets and realistic time to market.

Middleware
Distributed Object Middleware

Technologies
Servlets, EJB
Webservers: Weblogic & Websphere

OS Constructs
Servlets → HTTP
EJB uses RMI → Sockets

Paper# 28

Using XML Messaging for Wireless Middleware Communication

Summary
This paper basically tried to come up with a solution for Wireless Middleware that is efficient, reliable an secure as traditional middleware solutions such as RPC but still has the advantage of using XML. In this middleware the solution is arrived using two important concepts like Message Oriented Middleware (MOM) model for communication and XML for presentation. The implementation will allow a client and server to communicate using XML and SOAP messages. The middleware is being implemented in Java and is based on the JMS (java message service) framework. JMS will be customized to work with XML and SOAP messages and provide features that implement a reliable and secure channel for wireless clients.

Middleware
Wireless Middleware

Technologies
XML, SOAP, Java, JMS

OS Constructs
Message Queuing

Paper# 29

An XML Based, 3-tier Scheme for Integrating Heterogeneous Information Sources to the WWW

Summary
The extensive growth of the www currently experiencing the integration of various types of information sources to its platform. This paper presented a distributed architecture for the efficient retrieval of information residing in various types of databases like RDBMS or HTML collections. This architecture is based on the emerging XML standard as well

as ODBC, Java and HTTP. This architecture has five main components: Data sources, Agents, Integrators, Gateway Instances and XML-aware Java applets.

Middleware

Agent Based Middleware

Technologies

XML, Java, ODBC/OLEDB

OS Constructs

ODBC calls →RPC
Application calls → Sockets

Paper# 30

The Efficiency of XML as an Intermediate Data Representation for Wireless Middleware Communication

Summary

This paper presented an overview of different integration options and an introduction to wireless middleware. The use of XML as a possible solution intermediate data representation for communication was presented and the advantages and disadvantages of using XML were discussed. The main problem highlighted is the verbose nature of XML. A possible solution to this problem was proposed to use the compression to reduce message sizes. Two forms of compression were implemented. These are GZIP and WBXML. Initial testing showed that the prototype performed slower than a direct XML-RPC based solution but when either compression was used, it was quicker than a direct XML-RPC solution. The WBXML serialization was fast compare to GZIP.

Middleware

Wireless Middleware

Technologies

XML, WBXML, SOAP

OS Constructs

HTTP messaging

Paper# 31

Implementation and Performance of MidART

Summary

Addresses the problem of middleware design to support high speed network based distributed real-time applications. The class of applications it deals with are those in which humans need to interact with instruments and devices in a networked environment through computer-based interfaces. Control and monitoring systems are good examples.

Technologies

C++, MialSlot API (For message queuing)

Middleware

Message Oriented Middleware

OS Constructs

IPC (by using Mailslots)

Paper# 32

MQSeries Version 5 The Next Generation

Summary

This paper is intended to explain the features of the new version of MQSeries. This new version is stepped up to the IBM software server standards for product installation. These changes bring MQSeries into closer integration with IBM's DB2 database, Transaction server making it simpler to include these products together in an integrated solution.

Technologies

It supports CICS on sun Solaris

Supports tighter integration towards RDBMS using DMRC (Database message resource coordination)

Supports C++, IBM COBOL, Java

Middleware

Message Oriented Middleware

OS Constructs

RPC (to Put Messages to Queue and Get Messages from Queue)

Paper# 33

JMS – CORBA Notifications Service Interworking

Summary

The goal of PrimsmTech's interworking solution is to provide within the OpenFusion product suite a notification service-JMS bridge that can link EJBs deployed into any application server, whether it is IIOP-Compliant or uses other object transport protocols. This goal ensures that the solution is generic. The OpenFusion NS-JMS Bridge is compliant with the latest OMG Notification /Java Message Service.

Technologies
Java, JMS, CORBA

Middleware
Message Oriented Middleware

OS Constructs
Sockets

Paper# 34

Markup Meets Middleware

Summary
The main contribution of this paper is the discussion of an example (Trading system)of a successful combination of distribution middleware and markup languages that facilitates system integration. The architecture of the Trading system meets two main requirements: 1) Reliability transfer trading data between the distributed system components and 2) Resolve the heterogeneity of the data that is produced or expected by different trading system components.

Technologies
XML, CORBA

Middleware Type
Distribution middleware

OS Constructs
Sockets (If CORBA implemented in Java/RMI) or RPC (If CORBA implemented in C++)

Paper# 35

Exploiting Reflection in Mobile Computing Middleware

Summary
CARISMA is a project carried out at University of London, is a middleware that uses both structural and behavioral reflection to enable context-aware interactions between mobile applications. The middleware is in-charge of maintaining a valid representation of the execution context, by directly interacting with the underlying network operating system. Depending on the current context, applications may require the middleware to behave in specific ways.

Technologies
Java, XML

Middleware

Reflective Middleware

OS Constructs

Sockets

Paper# T6

Service Discovery Protocol Interoperability in the Mobile Environment –ReMMoC

Summary

The ReMMoC project being carried out at Lancaster University in collaboration with Lucent Technologies, is examining the use of reflection and component technology to overcome the problems of heterogeneous middleware technology in the mobile environment. The platform is constructed out of component frameworks, which consist of a configuration of components whose structure, can be altered via a meta-object protocol. The current platform consists of two key component frameworks for binding and service discovery.

Technologies

SOAP, COM, XML

Middleware

Reflective Middleware

OS Constructs

RPC

Paper# 37

An Efficient Component Model For the Construction Of Adaptive Middleware

Summary

OpenCOM is a lightweight and efficient in-process component model, built on the top of Microsofts COM. OpenCOM consists of the following: 1) the binary level interoperability standard 2) Microsoft's IDL 3) COM's GUID and 4) IUnKnown interface. This designed specifically for the development of middleware platforms. In other words, this exploits a component model for the construction of the middleware platform itself, which in turn provides an enhanced component model to application developers.

Technologies

COM, C++

Middleware

Reflective & Adaptive Middleware

OS Constructs

RPC

Paper# 38

XML Dataspaces for Mobile Agent Coordination

Summary

This paper presents programmable coordination architecture (MARS-X) for internet applications based on mobile agents. In this agents coordinate – both with each other and with their current execution environment – through programmable XML Dataspaces associated to each execution environment and accessed by agents, as if they were tuple spaces. This can provide several advantages in mobile agent applications, combining a high-degree of interoperability with a high-degree of interaction uncoupling.

Technologies

Java, XML Dataspaces

Middleware

Agent Based or Adaptive Middleware

OS Constructs

Sockets

Paper# 39

A Survey on Coordination Middleware for XML-Centric Applications - Displets

Summary

This paper analyzes several middleware systems proposed and implemented, to different extent and with different architectural solutions, aim at providing a coordination framework for a world of XML document agents. First one is Displets. The basic idea of the Dipslets approach is to provide an active document environment, where XML documents can be enriched with application-specific behaviour in order to say let them effectively rendered or transferred over a network. Displets are software modules that are attached to an XML document and activated when some pre-declared tags are parsed during the manipulation of documents.

Technologies
Java, XML

Middleware
Agent based or Coordination Middleware

OS Constructs
RPC

Paper# 40

A Survey on Coordination Middleware for XML-Centric Applications – XMLSpaces

Summary
XMLSpaces is an extension to the Linda model which serves as middleware for XML, In XMLSPaces, XML documents are fields within the coordination space. Thus, ordinary tuples are supported in terms of simple documents, while complex XML documents can be simply perceived as simple tuples with a single field (root of the document). The coordinator service could be implemented in some language running on the Java Virtual Machine

Technologies
Java, XML

Middleware
Agent based or Coordination Middleware

OS Constructs
RPC

Paper# 41

An XML Based Middleware for Peer-to-Peer Computing - XMIDDLE

Summary
XMIDDLE is an example of reflective middleware. It supports building mobile applications that use both replication and reconciliation over ad-hoc networks. It enables transparent sharing of XML documents across heterogeneous mobile hosts, allowing online and offline access to data.

Technologies
XML, XPath & Java

Middleware

Mobile, Data sharing, Reflective middleware

OS Constructs

Sockets

Paper# 42

A Lightweight XML-based Middleware Architecture

Summary

This paper presents a lightweight XML-based middleware for component communication with adaptation as an integral part. Connectors and their descriptions i.e document types are generated from the specifications. It uses static type information on the provided and required data to serialize and de-serialize the XML representations of services and data. XSLT specifications define the necessary transformations.

Technologies

XML, XSLT, XML Parser

Middleware

XML Based Middleware

OS Constructs

RPC or Sockets (Depends on Connector implementation)

Paper# 43

Rapid Application Development of Middleware Components by Using XML

Summary

Due to the close coupling of middleware and application design, changes and modifications of the middleware have always direct consequences on the design of applications. Each time new architecture is introduces the developers have to investigate how to integrate this with the middleware technologies. And if there are any specific changes in the middleware, they have to be implemented in the applications. This paper discuses the concept of generic XML object representations (OML) and middleware generators, which provides a loosely coupled integration environment which enables the rapid development of distributed applications. The automated generation of proprietary middleware components paves the ground for the development of applications in a heterogeneous interconnected system.

Technologies

XML, Java, RMI, COM

Middleware
XML Based Middleware

OS Constructs
Sockets

Paper# 44

JMS and Web Services Two Complimentary Standards

Summary
JMS and Web services are technologies that complement each other well. Web services will give us universal access, interoperability, independent of vendors' product. If properly used JMS add to Web Services a critically needed level of reliability and an improved way of handling user interactions. JMS as a transport protocol for SOAP calls is useful in specific set-ups only, but will not lead to a good solution in general, due to the complexity and vendor-independence introduced by the JMS binding.

Technologies
JMS, SOAP, Web Services

Middleware
Message Oriented Middleware

OS Constructs
HTTP calls

Paper# 45

Enterprise JavaBeans: The Value Proposition

Summary
Sun Microsystems came up with EJB, shortly after its creation of the Java language as a general-purpose development language, to extend Java to work in enterprise environments. To meet this Sun developed some J2EE as set of specifications. Enterprise JavaBeans provide a scalable, transactional, and multi-user secure environment for developing applications. These applications may be written once and then deployed any platform that supports EJB specification. This paper outlined some of the benefits of EJBs as well as some of the problems that have kept EJBs from fulfilling their potential in the market.

Technologies
EJB, J2EE

Middleware
Distributed Object Middleware

OS Constructs
Sockets

Paper # 46

Datasoft ODBCExpress White Paper (Programming ODBC: An Introduction)

Summary
The goal of this paper is to provide Delphi and C++Builder users with a clear, high-level understanding of the ODBC architecture. ODBCExpress wraps the ODBC API into a set of easy-to-use classes for the Delphi and C++Builder, so that user don't even have to know much about the low-level ODBCAPI to fully make use of ODBC in the applications.

Technologies
ODBC, SQL

Middleware
Database Middleware

OS Constructs
RPC

Paper# 47

Middleware for Universal Data access

Summary
Attunity Connect software is a new generation, standard based middleware solution from Attunity, Inc. With this multi tiered data access and integration solution, HP nonstop system users can access and manipulate data from diverse data access, residing on both local and remote platforms and across internet and intranets.

Technologies
C, SQL

Middleware
Database Middleware

OS Constructs
TCP/IP

Paper# 48

DCOM and CORBA side by Side, Step by Step, and Layer by Layer

Summary

This paper explains the DCOM and CORBA layer by layer and step by step. They both provide the distributed objects infrastructure for transparent activations and accessing of remote objects. DCOM supports objects with multiple interfaces and provides a standard QueryInterface() method to navigate among the interfaces. Both DCOM and CORBA frameworks provide client-server type of communications. To request a service, a client invokes a method implemented by a remote object, which acts as the server in the client-server model. In both DCOM and CORBA the interactions between a client process and an object server are implemented as object-oriented RPC style of communications.

Technologies

DCOM, CORBA, C++, Java, RMI

Middleware

Distributed Object

OS Constructs

DCOM → RPC
CORBA → Sockets (If implemented in RMI)
CORBA → RPC (If implemented in C++)

Paper# 49

Middleware Object Query Processing with Deferred Updates and Autonomous Sources

Summary

This paper presented a query processing algorithm called DECAF for use in middleware object query systems that use an object cache and in which updates to the underlying databases that don't go through the object cache are supported. DECAF query results are consistent with updates performed by such transactions, its results are also consistent with any deferred updates that are present in the object cache but not yet committed at the database server. The DECAF algorithm attempts to push down query predicates to the underlying DBMS's to take advantage of query processing capabilities of these systems and to reduce the amount of data transferred from the systems to the object cache.

Technologies

ADO, SQL

Middleware

Query Processing Middleware

OS Constructs

RPC

Paper# 50

MoCA: A Middleware for Developing Collaborative Applications for Mobile Users

Summary

MoCA (mobile collaboration architecture) is a middleware for developing context-processing services and context-sensitive applications for mobile collaboration. The work on this architecture is part of a wider project that aims to experiment with new firms of mobile collaboration and implement a flexible and extensive service-based environment for developing collaborative applications for infrastructured mobile networks. MoCA infrastructure consists of client and server APIs, basic services supporting collaborative applications, and a framework for implementing application proxies.

Middleware

Collaborative Middleware/Wireless Middleware

Technologies

Java

OS Constructs

Subscribe/public interfaces or Sockets

Paper# 51

Ganymed: Scalable Replication for Transactional Web Applications

Summary

Gynamed is as database replication middleware intended to provide scalability without sacrificing consistency and avoiding the limitations of existing approaches. This paper has presented a novel algorithm for the replication of databases at the middleware level as well as the design, implementation and evaluation of a replication platform based on this algorithm. Although the algorithm may appear to be conceptually simple it combines subtle insights in how real databases work and several optimizations that make it extremely efficient. The resulting system is light weight and avoids the limitations of existing solutions. The main idea is to use a novel transaction scheduling algorithm that separates update and read-only transactions. Transactions can be submitted to Ganymed through a special JDBC driver.

Middleware

Database Replication Middleware

Technologies
Java, JDBC

OS Constructs
Sockets

Paper# 52

iOverlay: A Lightweight Middleware Infrastructure for Overlay Application Implementations

Summary
In this paper the authors presented iOverlay, as a lightweight and high-performance middleware infrastructure that addresses problems like: failure detections and reactions, debugging and monitoring facilities, measurement of QoS metrics such as loss rates and per-link throughput. Application deployment and terminations, virtualzing distributed nodes, emulating resource bottlenecks and asymmetric network connections at the application level. iOverlay addresses these problems in a novel way by providing clean, well documented layers of middleware components. The interface between the iOverly and it's applications is designed to maximize the usefulness of iOvelay and minimize the programming burden of application developers.

Middleware
Distributed Object Middleware

Technologies
C# on Windows
C++ on Linux

OS Constructs
Sockets

Paper# 53

SyD: A Middleware Testbed for Collaborative Applications over Small Heterogeneous Devices and Data Stores

Summary
Existing middleware technologies like JXTA, .NET, J2ME require too many a-hoc techniques as well a cumbersome and time-consuming programming. This paper presented a System on Mobile Devices (SyD) middleware has a modular architecture that makes such application development very systematic and streamlined. The architecture supports transactions over mobile data stores, with a range of remote group invocation options and embedded interdependencies among such data store objects. The architecture

also provides a persistent uniform object view, group transaction with QoS specifications and XML vocabulary for inter-device communication.

Middleware

Mobile Middleware

Technologies

Java, Oracle, XML

OS Constructs

Remote Calls → TCP
Local Calls → RMI → Sockets

Paper# 54

The RUNES Middleware: A Reconfigurable Component-Based Approach to Networked Embedded Systems

Summary

This paper has presented the RUNES (Reconfgirable, Ubiquitous, Network Embedded Systems) approach to the development of software for networked embedded systems. This employs a uniform "Software Component" abstraction, which can be variously realized in various types of devices. RUNES approach to middleware is to build the middleware in terms of a well-defined language component model, which is supported by a minimal nature runtime API. The required heterogeneous realization of the component model in various types of devices is achieved by providing different implementations of the runtime API, and by implementing components themselves in various ways.

Middleware

Adaptive Middleware

Technologies

Java, Linux shared Objects

OS Constructs

Publish/Subscribe

Paper# 55

Portable and Efficient Distributed Threads for Java

Summary

The Existing Java middleware's RMI or CORBA do not support thread coordination over the network: synchronizing on remote objects does not work correctly and thread identity is not preserved for executions spanning multiple machines. This paper presented an approach that works with an unmodified middleware implementation, yet doesn't impose execution over head. The technique implemented in the context of J-Orchestra system, which is an automatic partitioning system for Java programs: given a Java application and under user guidance, it will split the application into parts that execute on different machines.

Middleware

Distributed Object Middleware

Technologies

Java, RMI

OS Constructs

Sockets

Paper# 56

A Middleware Infrastructure for Active Spaces

Summary

This paper presented a meta-operating system named Gaia built as a distributed middleware infrastructure that coordinates software entities and heterogeneous networked devices contained in a physical space. Gaia is designed to support the development and execution of portable applications for active spaces, programmable ubiquitous computing environments in which users interact with several devices and services simultaneously.

Middleware

Distributed Object Middleware

Technologies

CORBA

OS Constructs

RPC

Paper# 57

MiddleWhere: A Middleware for Location Awareness in Ubiquitous Computing Applications

Summary

This paper presented a distributed middleware infrastructure named MiddleWhere, for location that separates applications from location detection technologies and facilitates the incorporation of additional location technologies on the fly as they become available.

MiddleWhere uses probabilistic reasoning techniques to resolve conflicts and deduce the location of people given different sensor data. Besides, it allows applications to determine various kinds of spatial relationships between mobile objects and their environment, which is key in enabling a strong coupling between physical and virtual world, as emphasized by ubiquitous computing. MiddleWhere uses layered architecture for collection sensor information, representing it in spatial database and reasoning about it. Location sensors send information to spatial database through the MiddleWhere system.

Middleware

Distributed Object Middleware

Technologies

CORBA, PostgreSQL object-relational database

OS Constructs

RPC

Paper# 58

Java 2 Enterprise Edition (J2EE) Verses The .NET Platform Two Versions for eBusiness

Summary

This paper presented the comparison of two competing visions for eCommerce architectures. Sun's J2EE is based on a family of specifications that can be implemented by many vendors. It is open in the sense that nay company can license and implement the technology, but closed in the sense that it is controlled by a single vendor, and a self contained architectural island with very limited ability to interact outside of itself. The disadvantage is the choice of the platform dictates the use of a single programming language, and a programming language that is not well suited for most businesses and the major advantage is that the most of the J2EE vendors offer OS portability.

On the other hand Microsoft's .NET platform is family of products rather than specifications. The major disadvantage of this approach is that it is limited to windows, means applications written for the .NET platform can only be run on .NET platforms, but there are many advantages like cost of development is low, cost of running is low, greater scalability and stronger interoperability.

Middleware

Distributed Object Middleware, eCommerce Middleware

Technologies

.Net framework:	.Net, COM+, MSMQ, ADSI, DCOM, SOAP, ASP.NET,, ADO.NET
J2EE framework:	J2EE, RMI, IIOP, EJB, JMS, JNDI, JDBC

OS Constructs

.Net Framework:	RPC/Message Queuing
J2EE Framework:	Sockets/Message Queuing

Paper# 59

Understanding Performance Issues in Component-Oriented Distributed Applications: The COMPAS framework

Summary

In order to reduce the development costs for developing enterprise applications most of the times the developers use the commercial off-the-shelf (COTS) components. Due to the inherent complexity of such components, it is often difficult to model and predict the performance of the resulting application. This paper proposed a framework called COMPAS (Component Performance Assurance Solutions) that uses three interrelated modules to help developers understand and predict performance problems in component-based applications. Another goal of the framework is deriving QoS parameters for COTS components, relevant to the application being built, so that different COTS components with the same functionality from different vendors can be compared.

Middleware

Distributed Object Middleware

Technologies

Java, EJB, J2EE, JMX (Java Management Extensions), UML

OS Constructs

Sockets

Paper# 60

Autonomic Distributed Adaptive Middleware (ADAM)

Summary

Context aware middleware has to provide uniform development method for applications with balance between context awareness and transparency. To support ubiquitous application developers, ADAM middleware collects required context information; triggers middleware reconfiguration by well defined context inference and provides related services. In ADAM inference engine is employed to provide application with proactive service, network and system reconfiguration. This paper described ADAM structure and some scenario-based demonstrations.

Middleware

Adaptive Middleware

Technologies

Java, CORBA, RMI

OS Constructs

Sockets

Paper# 61

Thread Transparency in Information Flow Middleware

Summary

Existing middleware technologies is based on control-flow centric interaction models such as remote method invocations, poorly matching the structure of applications that process continuous information flows. This paper proposed middleware framework called Infopipe as a high-level abstraction for information flows, and this framework supports this abstraction. Infopipes transparently handle complexities associated with control flow and multi-threading. The platform is built on a message-based user-level thread package implemented in C++. Each thread consists of a code function and a queue for incoming messages. Unlike conventional threads, the code function is not called at thread creation time but each time a message is received.

Middleware

Information-flow Middleware

Technologies

C++

OS Constructs

Messaging (Push/Pull)

Paper# 62

Reliability of Composed Web Services from Object Transactions to Web Transactions

Summary

In this paper authors proposed an approach to build practical infrastructure support for loose transaction contracts. The goal of the infrastructure is to automate some of that support, making it cheaper to build, maintain and integrate. One service might not care how other service implements the contract, but the enterprise providing service A probably using its distributed object platform as the underpinning of its Web Service and service A will typically delegate work to smaller-grained distributed objects in the distributed system that underlies the web service. Authors identified research issues in the development of system infrastructure support for reliability of composed web services.

They investigated the suitable and applicability of object transactions, a common and proven approach to reliability in object systems, to web environments. The communication between web services however will not be used on the distributed object communication model of CORBA or J2EE. Rather an existing web services standards define, interactions across the web will be SOAP/XML message-based, and instead of a remote CORBA IDL or Java Interface, a WSDL specification is required.

Middleware

Web-Services Middleware

Technologies

CORBA/J2EE, SOAP/WSDL

OS Constructs

HTTP (SOAP Messages)

Paper# 63

RSCA: Middleware Supporting Dynamic Reconfiguration of Embedded Software on the Distributed URC Robot Platform

Summary

In this paper, the authors described RSCA Architecture as an integrated middleware, which supports the dynamic deployment of embedded software on the URC, distributed robot platform. Specifically they described the URC hardware and the application software architecture, and the necessity and functionalities of the RSCA standard operating environment. This RSCA standard operating environment is comprised of the RTOS, RT-CORBA distribution middleware, and the deployment middleware. The distribution middleware provides an abstraction layer that hides the heterogeneity of the distributed nodes in the URC robot and hides the decentralization of the system, there by making the distributed application components able to interact with each other flexibly. Also RT-CORBA distribution middleware supports the software component model for the distributed component-oriented computing for robot applications.

Middleware

Distribution Middleware

Technologies

CORBA

OS Constructs

RPC (ORB, IIOP)

Paper# 64

NaradhaBrokering: A Distributed Middleware Framework and Architecture for Enabling Durable Peer-to-Peer Grids

Summary

This paper has presented an extensible messaging framework which would be appropriate to host P2P grids. The test results demonstrated that the framework can indeed be deployed for both synchronous and asynchronous applications while incorporating performance-functionality trade-offs for different scenarios. This framework is incorporated with as an XML matching engine within the distributed brokering framework. This allows us to facilitate richer discovery mechanisms. Trade-offs in performance versus functionality inherent in such matching engines is a critical area that needs to be researched further. A P2P like discovery strategy within such a dynamic group combined with NaradhaBrokering's JMS mode between groups seems attractive.

Middleware

Messaging Middleware

Technologies

XML, JMS, JXTA

OS Constructs

Messaging (Publish/Subscribe)

Paper# 65

Adaptable Architectural Middleware for Programming-in-the-Small-and-Many

Summary

This paper provides a description and evaluation of a middleware intended to support software architecture-based development in the Prism (Programming-in-the-Small-and-Many) setting. The Middleware, called Prism-MW, provides highly efficient and scalable implementation-level support for the key aspects of Prism application architectures. Prism-MW is easily extensible to support different application requirements suitable for the Prism setting. Prism-MW is accompanied with design, analysis, deployment, and run time monitoring tool support. It has been applied in number of applications and used as an educational tool in a graduate level embedded by a major industrial organization for use in one of their key distributed embedded systems.

Middleware

Adaptable Middleware

Technologies
Java, C++, Embedded Visual C++

OS Constructs
Sockets, IPC

Paper# 66

A Flexible Middleware for Multimedia Communication: Design, Implementation, and Experience

Summary
Emerging multimedia applications require various communication features to be integrated and supported efficiently, which traditionally have been considered separately. For real world applications and integrated solution for communication middleware has to provide security and multicasting functionality in addition to multimedia services. The proposed middleware Da CaPo++ provides multimedia support within end –systems, which is adaptable to application needs. This concept is applicable to standard communication processing environments. Da CaPo++ integrates many aspects of research results obtained so far. The Da CaPo++ middleware demonstrates within a powerful and efficient system that a general-purpose end-system middleware for multimedia support is operational and interoperable with other end-systems applying Da CaPo++ as their choice of middleware.

Middleware
Multimedia Middleware

Technologies
C, C++

OS Constructs
IPC

Paper# 67

CORBA Implementation for Interface Software Agents with Virtual Test Bed

Summary
This paper discussed the preliminary implementation of the communication mechanism between a Multi-Agent system and the Virtual Test Bed (VTB) simulation package is discussed. This mechanism enables the interaction between a remote program and models in the simulation, so that the control of the program over the plant can be verified. The interface models can be placed on a power system schematic to extract and inject signals

during runtime. Several technologies, such as CORBA are used to make the model less dependent on the platform and languages of the client software programs.

Middleware

Agent Based Middleware

Technologies

CORBA, C++

OS Constructs

RPC

Paper# 68

Deep Middleware for the Divergent Grid

Summary

Next-generation Grid applications will be highly heterogeneous in nature, will run on many types of computers or devices, will operate within and across many heterogeneous network types, and must be explicitly configurable and runtime reconfigurable. This kind of grid environment is referred as "divergent grid". This paper proposed a "deep middleware" approach to meeting key requirements of the divergent grid. Deep middleware reaches down into the network to provide highly flexible network support that underpins a rich, extensible and reconfigurable set of application-level "interaction paradigms". In this middleware these facilities are encapsulated in two key component frameworks: the interaction and overlay frameworks.

Middleware

Grid Middleware

Technologies

C++, Java, JMS

OS Constructs

RPC, Publish/Subscribe

Paper# 69

Optimizing Java RMI Programs by Communication Restructuring

Summary

Java classes are modified at load time in order to intercept RMI calls as they occur. RMI calls are not executed immediately, but are delayed for as long as possible. When a dependence forces execution of the delayed calls, the aggregated calls are sent over to the remote server to be executed in one step. This paper presented an attempt to extend scope of run-time optimization to distributed systems. Conventional optimizing compilers, and optimizing virtual machines, focus on each node in a system individually. The authors have presented a prototype tool, which optimizes java RMI applications. The tool si based on powerful framework, essentially a virtual JVM, which allows the run-time system to re-order blocks of application code subject to data dependence metadata generated by static analysis. They used this to implement two optimizations of RMI applications: call aggregation, and call forwarding.

Middleware

Distributed Object Middleware

Technologies

Java, RMI

OS Constructs

Sockets

Paper# 70

The JBoss Extensible Server

Summary

JBoss in an open-ended middleware, in the sense that users can extend middleware services by dynamically deploying new components into a running server. This paper focuses on two major architectural parts of JBoss: its middleware component model, based on JMX model, and is meta-level architecture for generalized EJBs. The former requires a novel class-loading model, which JBoss implements. The later includes a powerful and flexible remote method invocation model, based on dynamic proxies, and relies on systematic usage of interceptors as aspect-oriented programming artifacts. On the top of JMX, JBoss introduces its own model for middleware components, centered on the concept of service component. The JBoss service component model extends and refines the JMX model to address some issues beyond the scope of JMX.

Middleware

Reflective Middleware

Technologies

JMX, EJB

OS Constructs

Sockets

3. SUMMARY AND CONCLUSIONS

The objective this paper was to investigate and collect various operating constructs needed for various middleware solutions. As I said in the introduction part I read hundreds of papers pertaining to following categories of middleware:

1. Database Middleware
2. Multimedia Middleware
3. Distributed Object Middleware
4. Reflective Middleware
5. Internet Middleware
6. QoS-Enabled Middleware
7. Configuration Middleware
8. Agent based Middleware
9. Web-based middleware
10. Wireless Middleware
11. Message Oriented Middleware
12. Adaptive Middleware
13. Coordination Middleware
14. Data sharing Middleware
15. XML based Middleware
16. Query Processing Middleware
17. Collaborative Middleware
18. Replication Middleware
19. Mobile Middleware
20. eCommerce Middleware
21. Information-Flow Middleware
22. Web-Services Middleware
23. Grid Middleware

In my investigation I have gathered following Operating systems constructs used by the papers I referred for my investigation:

1. RPC (Remote Procedure Calls)
2. Sockets (TCP Sockets, UDP Sockets, BSD Sockets etc)
3. ORB (Object request broker) and IIOP
4. Messaging
5. Message Queues
6. Publish / Subscribe
7. IPC
8. HTTP

I still believe that there are still some uncovered areas in this investigation.

4. REFERENCES

[1] Louis, D., Arun, I., Ilya, L., and Isabelle, R., A Middleware System Which Intelligently Caches Query Results

[2] Manual, R., Nick, R., MOCHAL A Self-Extensible Database Middlware System for Distributed Data Sources*

[3] Giedrius, S., Christian, S.J., Adaptable Query Optimization and Evaluation in Temporal Middleware

[4] Yoo, S.B., Kim, K.C., and Cha, K., A Middleware Implementation of Active Rules for ODBMS

[5] Geoff Coulson., A Configurable Multimedia Middleware Platform

[6] Claus Hofmann., A Multi tier framework for accessing distributed, hererogeneous spatial data in a federation based EIS

[7] Licia, C., Gordon, S.B., Cecilia, M., Exploiting Reflection in Mobile Computing Middleware

[8] Kim, K.H., Beck, D., Liu, J.Q., MiyaZaki, H., and Shokri, E.H., A CORBA Service Enabling Programmer-Friendly Object-Oriented Real-Time Distributed Computing

[9] Yamuna, K., Vishal, K., David, A.K., Craid, R., Joseph, P.L., and Richard, S., Integration of QoS-Enabled Distributed Object Computing Middleware for Developing Next-Generation Distributed Application

[10] Aykut, K., Nadir, A., Tufan, O, Alpay, D., A Combat Management System Based on CORBA

[11] Cynthia, M., An Object Infrastructure For Internet Middleware

[12] Carlos, O., David, L.L., Douglas, C.S., J. Russell Noseworthy., Applying a Scalable CORBA Event Service to Large-Scale Distributed Interactive Simulations

[13] Tsaoussidis, V., Badr, H., Na, L., Song, J., A CORBA-based Application Service Middleware Architecture and Implementation

[14] Nanbor, W., Michael, K., Douglas, CS., Applying Reflective Middleware to Optimize a QoS-enabled CORBA Component Model Implementation

[15] Gordon, S.B., Lynne, B., Valerie, I., Petr, T., Apostolos, Z., The Role of Software Architecture in Constraining Adaptation in Component-based Middleware platforms

[16] Valrie, I., Christos, K., and Apostolos, Z., Systematic Aid for Developing Middleware Architectures

[17] Scott, M.L., Frameworks for Component-Based Client/Server Computing

[18] Rui, S., Gordon, S.B., Eurico, C., A Reflective Component-Based & Architecture Aware Framework to Manage Architecture Composition

[19] Douglas, C.S., Frank, B., Patterns, Frameworks, and Middleware: Their Synergistic Relationships

[20] Andrew, T.C., Geoff, C., and Michael, E.K., Managing Complexity: Middleware Explained

[21] Christian, C., Burkhard, S., The Design of an Application Programming Interface for QoS-Enabled Multimedia Middleware

[22] Miroslav Benda., The Architecture Of Global Access

[23] Myron, H., Xuegao, A., Bing, Z., Yutao, H., OFTT: A Fault Tolerance Middleware Toolkit for Process Monitoring And Control Windows NT Applications

[24] Srinivasn, S., Loal, O., Chris, G., David, K., Tom, N., Rosy, C., Gene, M., Jannie, S., Global Change Master Directory: Object Oriented Active Asynchronous Transaction Management in a Federated Environment Using Data Agents

[25] Jian, Z., Minghui, Z., Quanyuan, W., An Agent Framework Based on Distributed Object

[26] Nikola, B.S., Developing Middleware for Web-aware Systems: Lessons Learned

[27] Max, D., Gerhad, B., and Michael, B., Java Servlets and Enterprise Java Beans in Enterprise Architectures: Friends or Foes Part II

[28] Wayne, S.H., and Prof. Ken, J.M., Using XML Messaging for Wireless Middleware Communication

[29] Petrou, C., Hadjiefthymiades, S., Matakos, D., An XML Based, 3-tier Scheme for Integrating Heterogeneous Information Sources to the WWW

[30] Wayne, H., and Kenneth, M., The Efficiency of XML as an Intermediate Data Representation for Wireless Middleware Communication

[31] Oscar, G., Chia, S., Ichiro, M., Morikazu, T., Implementation and Performance of MidART

[32] MQSeries Version 5 The Next Generation

[33] JMS – CORBA Notifications Service Interworking., A White Paper ny PrismTech Corporation

[34] Wlofgang, E., Walter, S., Anthony, F., Markup Meets Middleware[35] Licia, C., Gordon, S.B., Cecilia, M., Wolfgang, E., Paul, G., Exploiting Reflection in Mobile Computing Middleware

[36] Yerom-David, B., Valerie, I., Service Discovery Protocol Interoperability in the Mobile Environment

[37] Michael, C., Gordon, S.B., Geoff, C., and Nikos, P., An Efficient Component Model For the Construction Of Adaptive Middleware

[38] Giacomo, C., Letizia, L., Franco, Z., XML Dataspaces for Mobile Agent Coordination

[39] Paolo, C., Robert, T., Franco, Z., A Survey on Coordination Middleware for XML-Centric Applications – Displets

[40] Paolo, C., Robert, T., Franco, Z., A Survey on Coordination Middleware for XML-Centric Applications – XMLSpaces

[41] Cecilia, M., Licia, C., and Wolfgang, E., An XML Based Middleware for Peer-to-Peer Computing – XMIDDLE

[42] Welf lowe and Markus L. Noga., A Lightweight XML-based Middleware Architecture

[43] Gerd, Nusser., Ralf-Dieter, Schimkat., Rapid Application Development of Middleware Components by Using XML

[44] Martin, Erzberger., and Jorgen, Thelin., JMS and Web Services Two Complimentary Standards

[45] Enterprise JavaBeans: The Value Proposition

[46] Datasoft ODBCExpress White Paper (Programming ODBC: An Introduction)

[47] Middleware for Universal Data access

[48] P., Emerald, Chang., Yennum, Huang., Shalini, Yajnik., Deron, Liang., Joanne, C.S., Chung-Yih, Wang., DCOM and CORBA side by Side, Step by Step, and Layer by Layer

[49] Jerry, Kierman., Michael, J. Carey., Middleware Object Query Processing with Deferred Updates and Autonomous Sources

[50] Vanger, Scaramento., Markus, Enduler., Hana, K.R., Luciana, S.L., Kleder, G., Fernando, N.N., and Giulliano, A.B., MoCA: A Middleware for Developing Collaborative Applications for Mobile Users

[51] Chrystain, Plattner., and Gustao, Alonso., Ganymed: Scalable Replication for Transactional Web Applications

[52] Baochun, Li., Jiang, Guo., Mea, Wang., iOverlay: A Lightweight Middleware Infrastructure for Overlay Application Implementations

[53] Sushil, K.P., Vijay, M., Shamakant, B.N., Raj, S., Erdogan, D., Anu, B., Michael, W., Raghupathy, S., Bing, L., Janaka, B., Arthi, H., Wanxia, X., Praveen,M., Srilaxmi, M., Alex, Z., Yanqing, Z., Yi, P., and Saied, B., SyD: A Middleware Testbed for Collaborative Applications over Small Heterogeneous Devices and Data Stores

[54] Paolo, C. Geoff, C., Cecilia, M., Gian, P.P., Stefanos, Z., The RUNES Middleware: A Reconfigurable Component-Based Approach to Networked Embedded Systems

[55] Eli, T., and Yanis, S., Portable and Efficient Distributed Threads for Java

[56] Manual, R., Christopher, H., Renato, C., Anand, R., Roy, H.C., and Klara, N., A Middleware Infrastructure for Active Spaces

[57] Anand, R., Jalal, A., Shiva, C., Roy, C., M. Dennis Mickunas., MiddleWhere: A Middleware for Location Awareness in Ubiquitous Computing Applications

[58] Roger, Sesions., Java 2 Enterprise Edition (J2EE) Verses The .NET Platform Two Versions for eBusiness

[59] Adrian, M., John, M., Understanding Performance Issues in Component-Oriented Distributed Applications: The COMPAS framework

[60] Kwang-Hoon, C., Jai-Hoon, K., Young,-Bae, K., Hee, Y.Y., we-Duke, Cho., Autonomic Distributed Adaptive Middleware (ADAM)

[61] Rainer, Koster., Andrew, P.B., Jie, H., Jonathan, W., and Calton, P., Thread Transparency in Information Flow Middleware

[62] Thomas, M., Isabelle, R., Stefan, F., Reliability of Composed Web Services From Object Transactions to Web Transactions

[63] Jaesoo, L., Jiyong, P., Seunghyun, H., and Seongsoo, H., RSCA: Middleware Supporting Dynamic Reconfiguration of Embedded Software on the Distributed URC Robot Platform

[64] Shrideep, P., and Geoffrey, F., NaradhaBrokering: A Distributed Middleware Framework and Architecture for Enabling Durable Peer-to-Peer Grids

[65] Marija, Mikic-Rakic., Nenad, Medvidovic., Adaptable Architectural Middleware for Programming-in-the-Small-and-Many

[66] Burkhard, Stiller., Christina, Class., Marcel, Waldvogel., Germano, Caronni., Daniel, Bauer., A Flexible Middleware for Multimedia Communication: Design, Implementation, and Experience

[67] Gilberto, M., Li-Hsiang, S., David, A.C., CORBA Implementation for Interface Software Agents with Virtual Test Bed

[68] Paul, G., Geoff, C., Gordon, B., and Barry, P., Deep Middleware for the Divergent Grid

[69] Kwok, C.Y., and Paul, H.J.Kelly., Optimizing Java RMI Programs by Communication Restructuring

[70] Marc, F., and Francisco, R., The JBoss Extensible Server