CS 406

Tivoli Systems Project

Problem Statement

Creating a Persistent Data Store with Lotus Notes

August 18, 1998

Introduction

This document specifies a problem statement for developing a persistent object storage facility for use within Tivoli Systems' customer service software. This problem statement identifies the parameters defining the scope of the project for CS 406 students. These parameters include a description of existing architecture constraints affecting the persistent object storage facility, software design and development tools to be used on the project, and the general software engineering approach for implementing software that satisfies this problem statement.

Tivoli Systems Service Management Background

Tivoli Systems is a division of IBM that develops software for "systems management" and "application management". This management software allows users of Tivoli software to monitor and correct failures in the hardware and software systems they use to run their businesses. Typical customers use Tivoli software to maintain and maximize the availability (or "up time") of their key business systems. Such key business systems may include Internet servers, file servers, database servers, payroll processing systems, order entry systems, and billing systems.

The software architecture used by Tivoli management software for monitoring and correcting system failures involves the use of a central event processing server and software "agents" deployed on the machines and/or processes that are being managed. These agents notify the central event server of a system failure and the event server can then take action on the event. Actions taken by the event server can include displaying a log message, notifying a person via e-mail or pager, forwarding the event to another event server, or instructing the software agent to take a corrective action, such as restarting the process that failed.

Integrating with the systems management event server are several other Tivoli products. Like the central event processing server, each of these products is designed to help manage a company's information technology (IT) infrastructure. Included among these other Tivoli products are automated software distribution, job scheduling, network management, systems performance monitors, and help desk software.

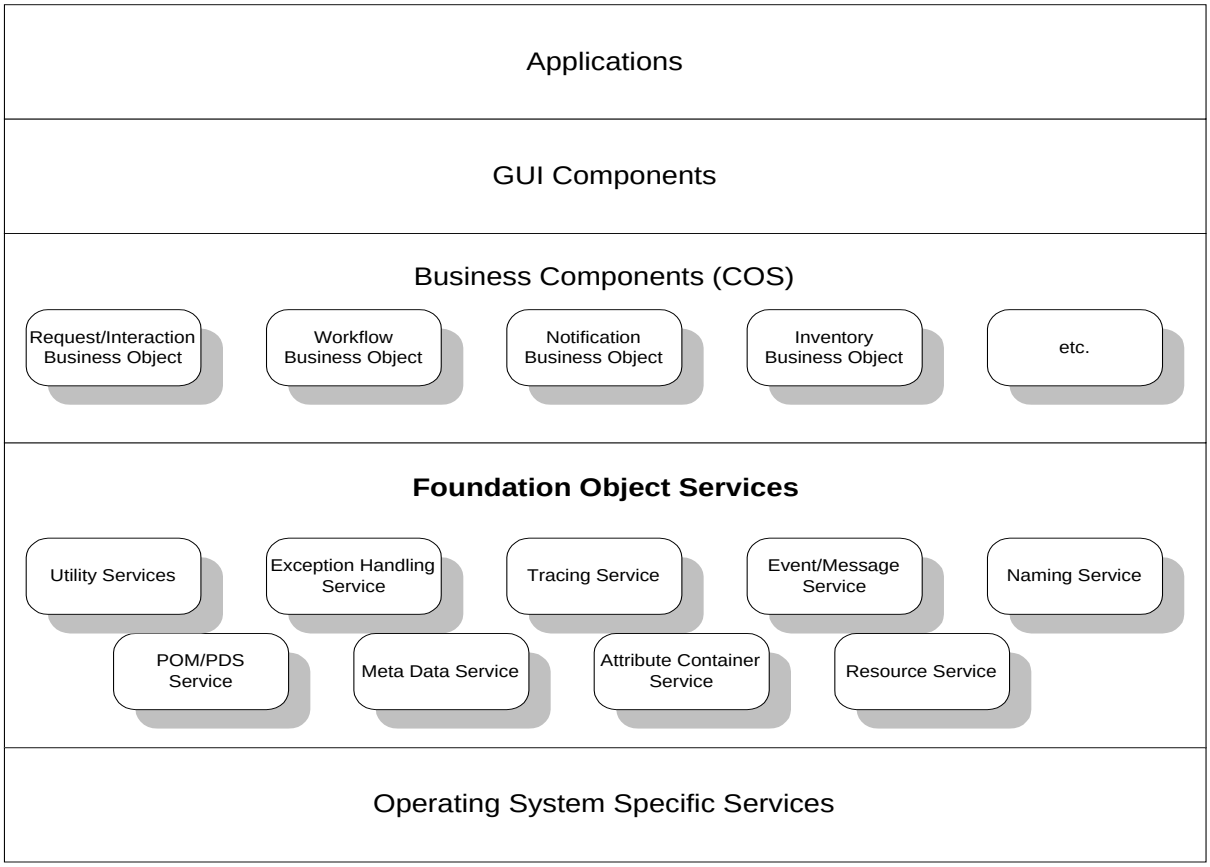
Tivoli's Indianapolis operation develops and supports the help desk products within the Tivoli IT management suite of products. Help desk software is used to operate a "call center" that responds to requests for service. Such requests for service come from users of IT resources within a company who, when they have a problem, call a central phone number or use a special e-mail address to report their problem to the call center. Sample requests for help that a user of IT resources might report include calling because their PC won't boot, calling because the user can't connect to a web site, or calling because the user can't connect to a network printer. The help desk is staffed by help desk analysts whose job it is to record the caller's problem, try to solve the problem by question and answer, looking up previous, similar problems for possible solutions, or by transferring the call to another analyst with more experience.

Tivoli Systems "Magellan" Project

Tivoli-Indianapolis is in the process of redesigning and rewriting its help desk software. The project to redesign and rewrite is code-named "Magellan". This Magellan project's design includes a set of business objects that define the intelligence of the help desk software and a set of graphical user interface (GUI) components that interact with the business objects. The help desk application used by a help desk analyst to record and solve a caller's problem is written using the graphical components.

The business objects are written in C++, while the graphical components and the help desk application are written in Java. The basic architectural layers of the Magellan help desk software are illustrated below.

Magellan Project : Architectural Layers



Magellan Persistent Object Services

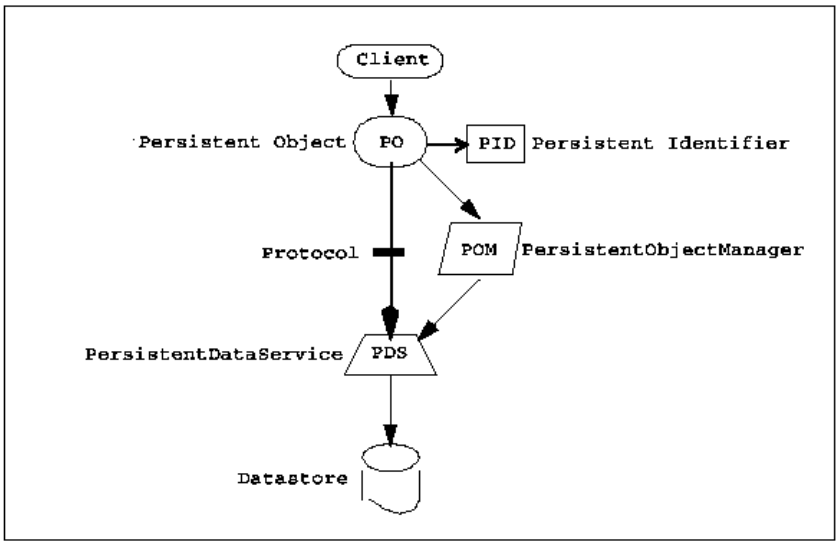
The specific problem for CS 406 is to design and implement a new Persistent Data Store (PDS) facility using Lotus Notes as the database.

The preceding diagram illustrates the four layers of Magellan – Applications, GUI Components, Business Components, and Foundation Object Services. The Foundation Object Services layer is the layer that provides basic utilities and services for the business objects (components). A key capability of the framework used by the business objects is storing objects persistently. This persistence mechanism allows the help desk to keep a permanent record of help desk data such as calls received by the help desk, solutions used to solve customer problems, and caller identification information (name, department, phone number, etc.).

The current persistent object services facility in Magellan is modeled after the Object Management Group (OMG) specification for object persistence. The persistent object services of OMG and Magellan specify use of three major components for implementing object persistence. These three major components are the Persistent Object (PO), the Persistent Object Manager (POM), and the Persistent Data Service (PDS).

A diagram illustrating these components and their interactions is given below.

OMG and Magellan Object Persistence



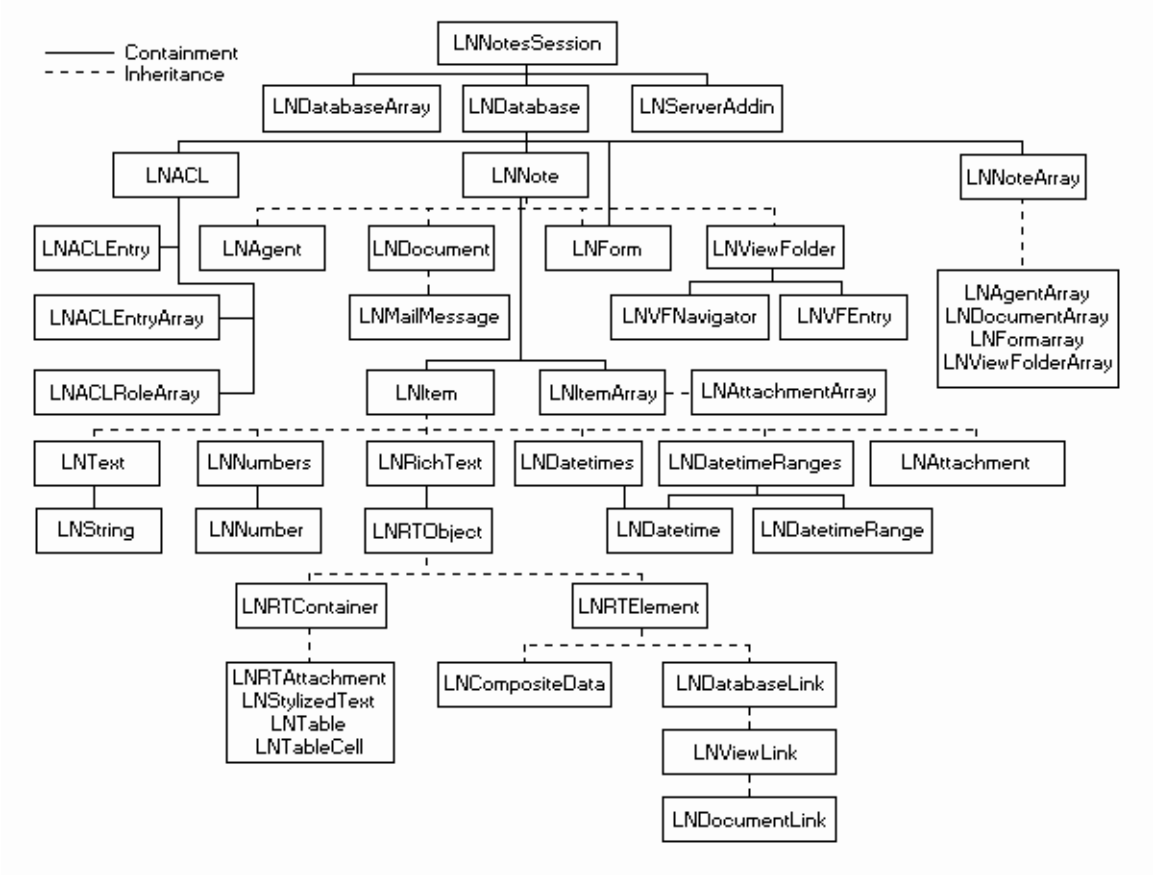
With this approach for object persistence, the Persistent Object communicates high-level requests to the Persistent Object Manager (POM) and, based upon an object's Persistent ID (PID), assigns and delegates the requested operation to a Persistent Data Service (PDS). This approach allows a consistent interface for object persistence through the POM with the POM acting as a "traffic cop" to direct the persistence operation to a specific PDS. The PDS implementation provide a standard set of interface methods, but are implemented in a data store-specific manner. For instance, a single POM can communicate with a PDS that stores to a relational database, to another PDS that stores to a flat file, and perhaps to another PDS that stores to an ISAM file structure.

For the Tivoli Magellan implementation of PDS, the following interface methods are defined on the PDS :

Store()	Store an object persistently
Restore()	Restore (read) an object from persistent storage
Delete()	Delete an object from persistent storage
GetPdsStatus()	Return status information about the PDS server
Select()	Issue a query to retrieve a list of objects from persistent storage
Fetch()	Retrieve objects from a list created by Select()
StartTrans()	Start a transaction on a PDS
CommitTrans()	Commit (store) all objects modified during a transaction
RollbackTrans()	Abort all object modifications involved in a transaction
Stop()	Shutdown the PDS server

Lotus Notes

Lotus Notes is a messaging (e-mail) product that includes a full-fledged database, among other features. Notes provides a C and a C++ interface to its internal database. The object model for the Notes Database C++ interface is shown below.



Project Approach

The goal of this Tivoli project is to analyze, design, and develop a PDS implementation that uses Lotus Notes as the persistent storage database. This project is a "real-world" project that will allow students to learn and apply a phased approach to software engineering. The phases to be applied on this project include analysis, design, implementation, testing, and support. Tools used in the analysis and design phase include the Rational Rose visual modeling tool, the Unified Modeling Language (UML). Tools used in the development, testing, and support phase include C++, Common Object Request Broker Architecture (CORBA), C++, and Lotus Notes.

Further Reading

www.omg.org – Object Management Group web-site
ftp:www.omg.org/pub/docs/formal/97-12-12.pdf - Object Management Group Persistent Object Service Specification
www.lotus.com/home.nsf/tabs/development – Lotus Notes developer web site