Formal Methods in Software Engineering

2011-12 project on

Software Architecture for Reactive Systems

Contents

1 Instructions

Each case presents a problem description and solution guidelines. The first constitutes the requirements for a system and the second prescribes architectural decisions.

You will be assigned one case; describe an architecture for a system that fulfills the requirements in the problem description and follows the solution guidelines. For the description choose the formalism(s) you consider is(are) better suited to represent and verify the required properties.

2 Cases

2.1 The Sea Buoy System

2.1.1 Problem Description

A Sea Buoy System manages sensors, a radio transmitter, a radio receiver, and an emergency switch installed on a free-floating buoy. The sensors collect air and water temperatures, wind

speed, and location data on a periodic basis. The radio must transmit in either of three situations: to periodically broadcast collected information, to send the last 24 hour data to a passing vessel that has requested it through the receiver, to broadcast an SOS signal in the case a sailor is able to reach the buoy and turns the emergency switch on. The SOS broadcast has priority over any on demand transmission which, in turn, is more important than periodic transmissions. The system architecture is expected to: allow the integration of these loosely-coupled activities; facilitate meeting priorities and time constraints; permit modifications, as components may be replaced, or new sensors or actuators may be included, or priorities or time constraints may be changed.

2.1.2 Solution Guidelines

The system should be resolved materializing the Blackboard Architectural Pattern. This pattern suits situations that require the integration of loosely-coupled and heterogeneous components that may be replaced over time. It prescribes two basic types of components: a central data structure - *blackboard*; and independent, loosely-coupled components - *knowledge source*. The blackboard represents the state of the system. It includes domain data to resolve the problem at hand, and control data to coordinate the execution of knowledge sources. Each knowledge source addresses a particular concern and can only communicate with the blackboard, adopting a star configuration with the blackboard as center.

2.2 XYZ News' Website

2.2.1 Problem Description

XYZ News is proud to be the reliable online news agency to first provide information in different formats to its audience. Their journalists arrive where news take place and prepare text, images and videos to be put online. However, their reputation has a price. Whenever an important event takes place, the website receives massive amounts of requests for a short period of time. XYZ applies two tactics to manage these peaks. The first is to dinamically add up to N web servers to the cluster and the second is to provide only text content in articles. Then, a server can be attending requests in either a full-media or a text-only mode. The first tactic is used until there are no more available servers. Then, the second is applied. XYZ board wants that every request is answered, and if possible, in less than n ($n > m$) miliseconds. They know that a server can respond in less than m miliseconds up to R simultaneous requests in full-media mode, and up to Q in the text-only one.

2.2.2 Solution Guidelines

The system should be resolved by composing Client Server and Pipes and Filters Patterns. A filter receives request from clients and decides which web server (another filter) will respond to it. Within each web server there are three filters: one filter to answer in full-media mode, another to answer in text-only mode, and a third to decide which one of them answers the request.

2.3 IPCall's System

2.3.1 Problem Description

IpCall provides their clients with a variety of communications services. Clients are assigned a phone number to make and receive phone calls. They can register their communication devices (internet phones, mobile phones, fixed lines, etc ...) and when a call is received, it is forwarded to all of these devices either in parallel or in sequence. In the later case, if the client has not answered the call after an amount of time, the call is forwarded to the next device in the list. The client can also configure a voice message for the calls he is not able to answer, and an email account to receive any voice message left by callers. IpCall has a website with other services. Clients can sign in and see a log with all the calls they have done and received. In addition, they can also initiate a call from the website by clicking a "Connect" button each of their contact entries has. The system is required to register all calls. No call can be disregarded or lost. When a call is initiated from the web, the system must call the two parties and connect them (provided they accept the call immediately) within the next T seconds after the web request was issued.

2.3.2 Solution Guidelines

The system should adopt a client-server architecture in which the server is logically organized into three Layers (see Figure 2.3.2). The first is the Presentation Layer, that deals with client interaction and comprises a Web Server interacting with Web Browsers and a Communication Server interacting with internet phones and fixed line ones. The second, the Business Logic Layer, is an Application Server that interacts with the two servers in the presentation layer and the third layer described next. The third is the Persistence Layer, which is a data repository.

2.4 Delfos' System

2.4.1 Problem Description

Delfos provides call center services to their clients: small and medium enterprises — callees, that need to answer calls from their respective clients — callers. A callee hires a number N of Delfos' agents that will be instructed into the callee's business in a way that they can answer any question in less than a minutes. Delfos provides a warranty that all calls will be attended with a wait no longer than w seconds (provided a maximal agreed call rate). All calls are stored in case the callee requires to audit them. The system receives calls and queue them until a scheduler process assigns it to an agent. The scheduler can assign calls in the queue following a round-robin or a FIFO basis. While the first policy checks which idle agent is next according to a cycle, the second takes the next agent in a queue of idle ones. Delfos' CTO also wants to be sure that no call is lost, and no call is dispatched to the wrong agent, i.e., an agent working for a callee A receives a call for a callee B.

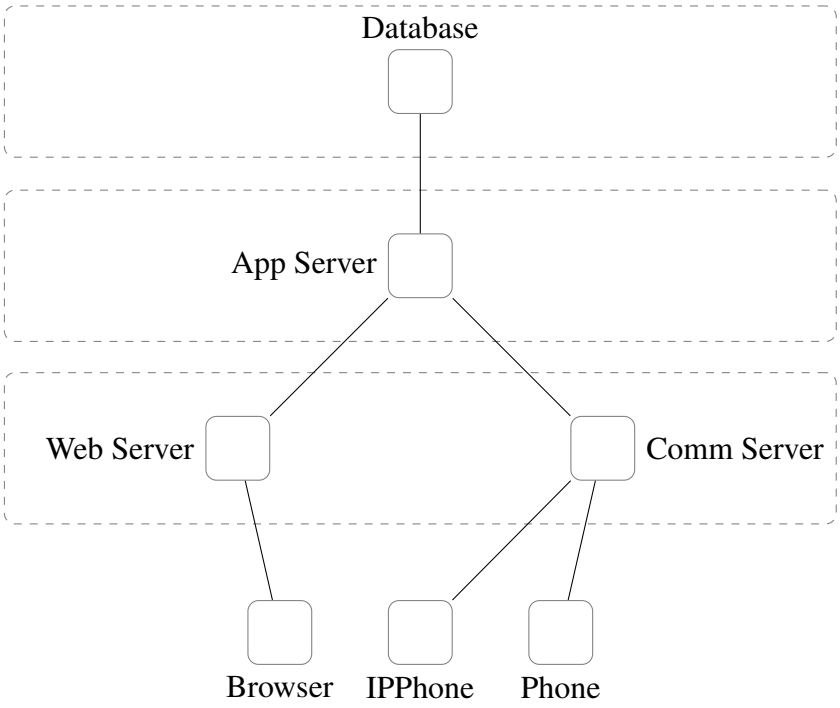

Figure 1: IPCall System

2.4.2 Solution Guidelines

The system should be resolved hierarchically composing materializations of the Pipes and Filters Pattern. A queue filter receives the calls and delivers them to a scheduler filter on demand. The scheduler subsystem is itself a pipe and filter system with one filter for the round-robin policy and another for the FIFO one. Once the scheduler has assigned the call an agent, it passes this information to a connect filter that connects the call to the agent. A call may finalyze or cut in any moment by the caller. In any case, and regardless of the filter where the call is, its details are sent to a persistence filter for storage.