Requirements Analysis Document

PAID Project

Network−Team

Informatik XII

WS 1998/1999

Technische Universität München

January 14, 1999

Revision History:

- Version R0.1 9/15/98 Robin Loh. Created
- Version R0.2 12/16/98 Martin Uhl. Modified for TUM
- Version R0.3 12/21/98 Martin Uhl. Added automatical paragraph numbering
- Version R1.0 01/11/99 Marko Werner

Preface:

This document addresses the requirements of the PAID system. The intended audience for this document are the designers and the clients of the project.

Target Audience:

Client, Developers

PAID Members:

Project Management:	Bernd Bruegge, Guenter Teubner
Team Coaches:	Ralph Acker, Stefan Riss, Ingo Schneider, Oliver Schnier, Anton Tichatschek, Marko Werner
Architecture Team:	Asa MacWilliams, Michael Luber
Authentication & Security Team:	Klaas Hermanns, Thomas Hertz, Guido Kraus, Gregor Schraegle, Tobias Weishaeupl, Alexander Zeilner
Database Team:	Osman Durrani, John Feist, Florian Klaschka, Johannes Schmid, Florian Schoenherr, Ender Tortop
Learning Team:	Burkhard Fischer, Juergen Knauth, Andreas Loehr, Marcus Toennis, Martin Uhl, Bernhard Zaun
Network & Event Service Team:	Henning Burdack, Joerg Dolak, Johannes Gramsch, Fabian Loschek, Dietmar Matzke, Christian Sandor
StarNetwork Integration & User Interface Team:	Daniel Stodden, Igor Chernyavskiy, Inaki Sainz de Murieta, Istvan Nagy, Stefan Krause, Stefan Oprea
Testbed Team:	Bekim Bajraktari, Bert van Heukelom, Florian Michahelles, Goetz Bock, Michael Winter, Sameer Hafez

MILESTONES

- 1/11/99 Release of RAD Template
- 1/14/99 Team RAD
- 1/18/99 Integerated RAD

Table of Contents

1 General Goals

The goals of the network subsystem are to reliably transmit messages and data between Daimler–Benz after–sales database servers and dealer machines running the PAID system. The information should be accurate and delivered in a timely manner. The communication between PAID–subsystems will also be provided.

All updates to the database should be pushed to dealers who are using the information using a multi–cast distribution policy.

2 Current System

Currently, the entire database is distributed using CD–ROM, microfiche, online, and paper methods to the various dealers.

Redistribution occurs on a monthly basis.

3 Proposed System

3.1 Overview

The PAID Network subsystem will be designed to provide a method of communication between a dealer and the central database, as well as providing communication between the other individual subsystems of PAID. Specifically, the Network subsystem will allow a remote user to request information from the central database, and to receive updated information from the central database.

3.2 Functional Requirements

The Network subsystem will implement both pull and push functionality. Pull functionality will allow a dealer (on a remote machine) to request the transferral of data from the central database. Push functionality will allow a central database to send updated information to a dealer. Also, the system will allow updates to be multi–cast to multiple dealers.

Transparently to the above mentioned functionality, the Network subsystem will provide on–the–fly compression (using a scheme such as adaptive compression) to reduce network load. Error checking will also be incorporated to ensure that data passed over the Network is uncorrupted. The Network subsystem will monitor server load and network responsivity to assure that information is routed in the most effective way possible.

3.3 Nonfunctional Requirements

Currently, the entire database is distributed using CD–ROM, microfiche, online, and paper methods to the various dealers.

3.3.1 User Interface and Human Factors

The Network subsystem will not be interacting with human users, instead the users of the Network subsystem will be interacting solely with other subsystems within PAID. The Network subsystem will expose an Application Programming Interface (API) via which these other subsystems will interact with the Network subsystem. Also, the internal workings of the Network subsystem will be transparent to the other subsystems.

3.3.2 Documentation

Documentation will be provided that covers the specifics of the API, in terms which can be understood by the developers of the other PAID subsystems. The documentation will also cover how the Network subsystem will react under specific conditions. We will also be using JavaDoc to generate documentation from the coded implementation. All documentation, TogetherJ projects, and code, and libraries shall be managed via CVS.

3.3.3 Hardware Consideration

The Network subsystem does not deal with specific machine−type constraints. Since the system will be implemented in Java, the code will be usable on a variety of machines. The type of connection used (Modem, Ethernet, ATM, ISDN, GSM, APS, etc) will have an affect on the network subsytem's operation, in that the Network subsystem will attempt to determine the speed of the connection, and adjust the network parameters (such as TCP/IP window size and compression methods) in order to make full use of the connection speed and to minimize congestion.

3.3.4 Performance Characteristics

The Network subsystem is expected to transmit at the highest speed allowable based on the connection between a remote machine and the central database. The estimated ratio of server to clients will be 1:150. In the event of a network overload, the network will use shared throughput and attempt to circumvent bottlenecks by transmitting packets along an alternate routes.

Response time will be dependant on how quickly congestion is discovered. We estimate 5−7 transactions during peak times and 2−3 transactions during off−peak times. By using shared throughput we will balance the server high bandwidth with vs. the dealer low bandwidth. We will be using adaptive compression to reduce network load.

3.3.5 Error Handling and Extreme Conditions

There are three exceptional situations that may arise. The first situation is network problems. In the case that the network becomes unreliable, any currently ongoing transmissions will be terminated, and both ends (the remote machine and the central database) will be informed of the interruption as well as the state of the transmission upon termination. The second possible situation is that the dealer (on a remote machine) may choose to terminate a transmission, in which case the central atabase will be informed of the situation and both ends will be informed of the state of the transmission at the time of termination. The third possibility is that the local program is terminated unexpectedly in the middle of a transfer (e.g. in case of power failure). In this case, the client would clear recieved information from its cache upon reboot and continue as if transfer never occured.

3.3.6 System Interfacing

The network subsystem functions as the lowest layer or core of the PAID project. All messages sent over the network will be passed through this subsystem. All interfacing to this subsystem will be done through the Network API (NAPI). However, there are two main subsystems that we expect to interface with NAPI in order to provide the basic functionality of PAID. We anticipate the event/learning subsystem to pass data for automatic server to client updates through the network, in the form of either a point to point or multi−cast message. We also expect the database subsystem to request and receive desired information from the server through the network subsystem. All interactions between these subsystems will take place on the CORBA bus through a Java interface.

3.3.7 Quality Issues

This subsystem must provide fast, portable, and reliable information transportation over the network for the PAID system. In order to maximize speed efficiency, the network subsystem will provide compression on the fly, be selective (multi−cast or point to point) to the amount of information transported (to prevent unnecessary lags), and optimize routing of the information (in terms of time). To ensure portability, the subsystem is fully implemented in Java. This means any hardware/operating system which has a Java Virtual Machine will be able to route information through PAID over the network.

3.3.8 System Modifications

As more dealers are added to the system, scalability issues might have to be considered if the network's response is being depreciated by an overload of responses to the server. We do not anticipate any modifications outside of this as we expect the network subsystem to be very stable. We achieve this by using Java as our language as it is cross−platform and in wide use.

3.3.9 Physical Environment

This is not an issue because we are using Java technology, which is independent from the actual hardware PAID is running on.

3.3.10 Security Issues

All user access to data will be controlled by authentication and database subsystems. Transport over the Daimler−Benz intranet and extranet needs no special security measures. Transport over the general internet needs to be encrypted to insure security.

Physical security is important for both the database servers and for local copies of the database on dealer machines.

3.3.11 Resource Issues

System installation will be performed along with the rest of the application. Little to no maintenance will be required for the network subsystem after the initial setup stage. Neither will any persistent data be externally accessible.

3.4 Constraints

System will be developed with Java (ver. 1.1.x). Object and CASE models will be created with Together/J. Source Control will be handled using CVS.

3.5 System Model

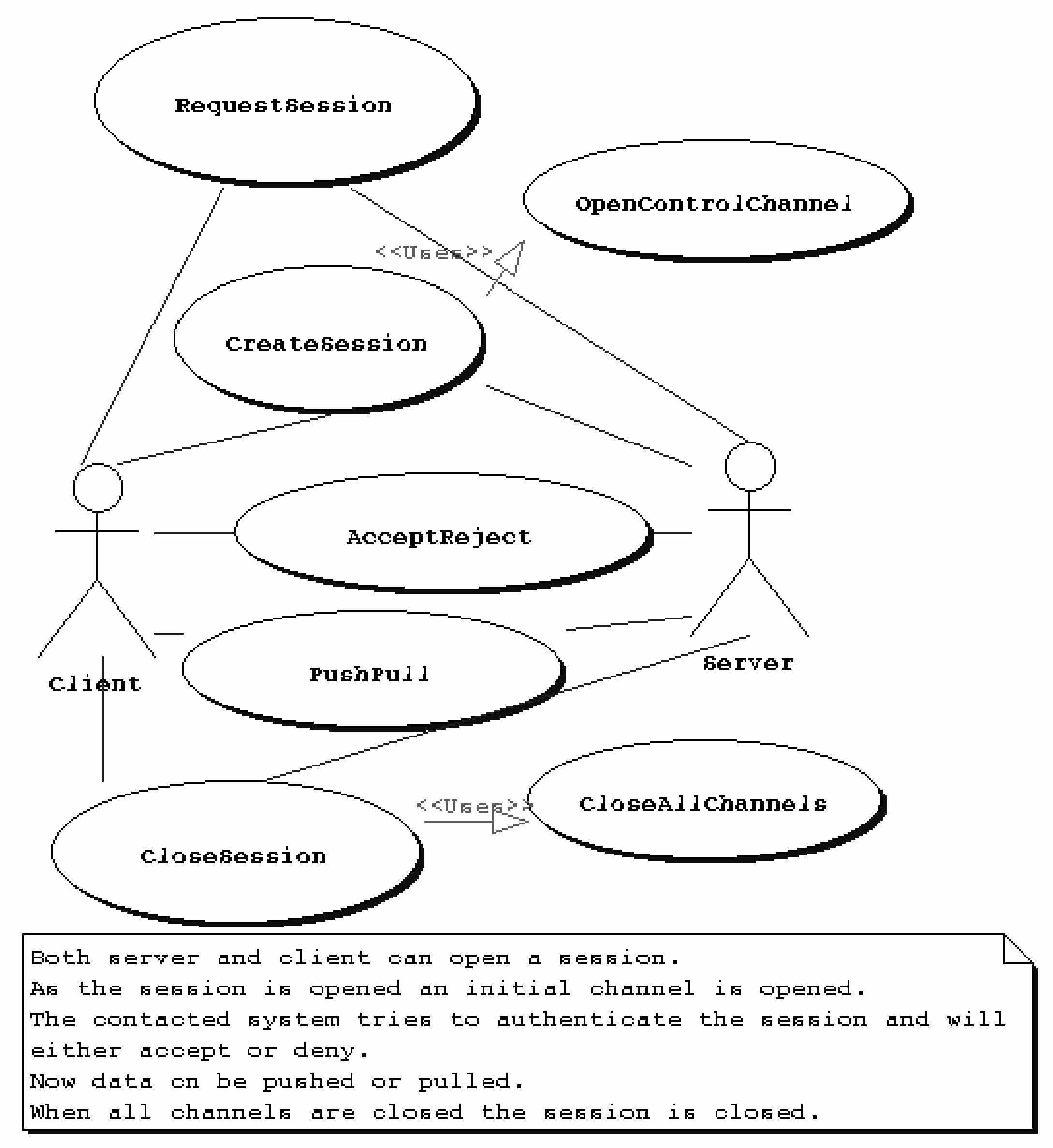
3.5.1 Scenarios

[Push] Mr. Benz has come out. with a new car model. He adds information about the new model to the central database in Germany. This new data needs to be transmitted to all dealers interested in this class of car. The central database tells the network to announce the availability of the new information to the relevant dealers. Bob (a dealer in San Francisco) receives the notification and requests, via the network subsystem, that the data be sent immediately. The network subsystem will be making use of the CORBA bus to remotely call procedures on the database server. The central database, upon receipt of the acceptance, now sends the updated information to Bob's database via the network. Jim (another dealer in Miami), is busy with clients and requests that he be notified later. The network sends the request to the central database, which will log this request and later send Jim another notification.

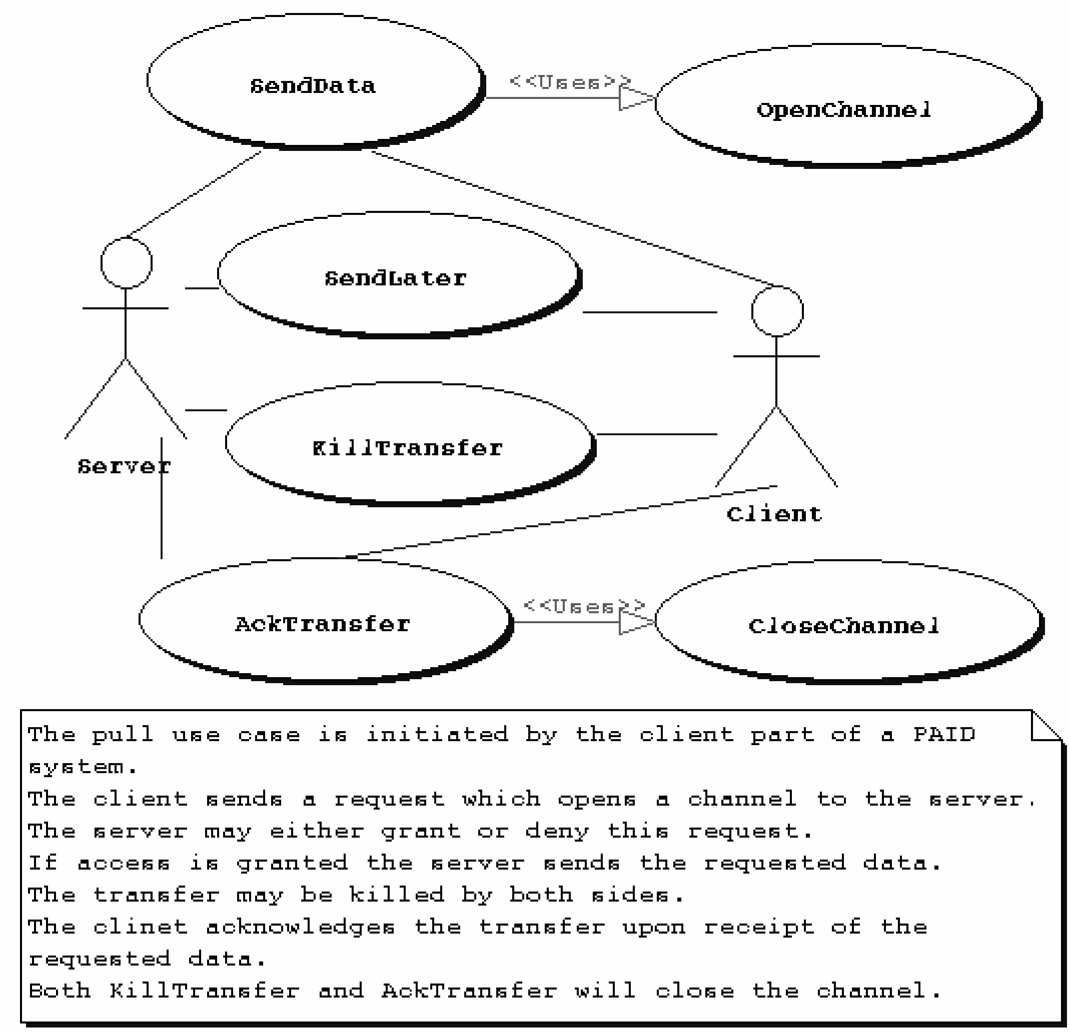
[Pull] Humphrey, a Deimler–Benz dealer in Edinburg, has decided to expand his model selection with a new S–class model. He logs into the PAID system, and fills out the a form to request information on the model. He clicks the submit button, and the his local database discovers that it doesn't contain the desired information, so it passes the request to the network subsystem. The network subsystem uses CORBA bus to request information from the central database. The central database will then use the network subsystem to transmit the request information back to the database to Humphrey's machine.

3.5.2 Use Case Models

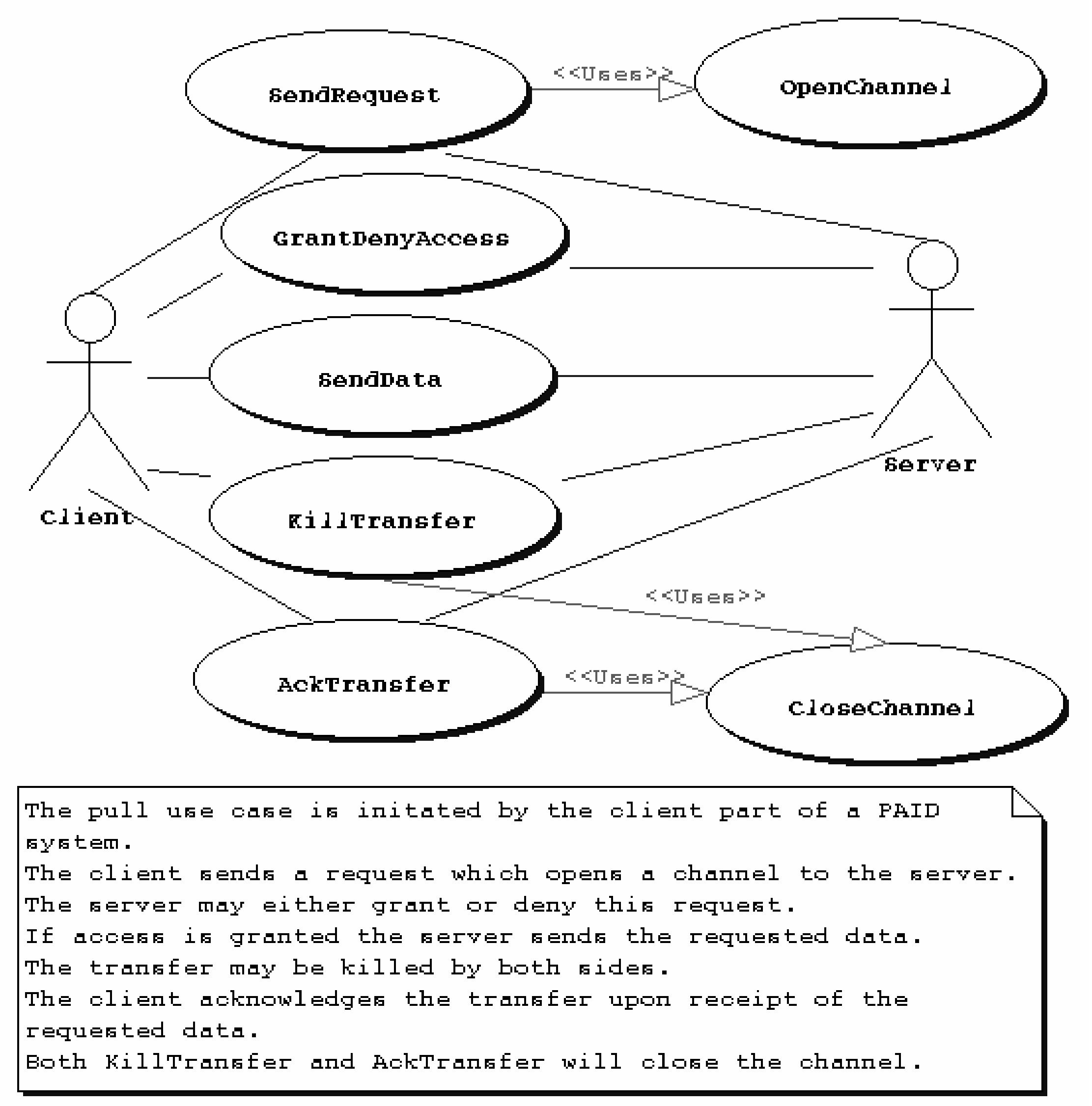
Session Use Case :



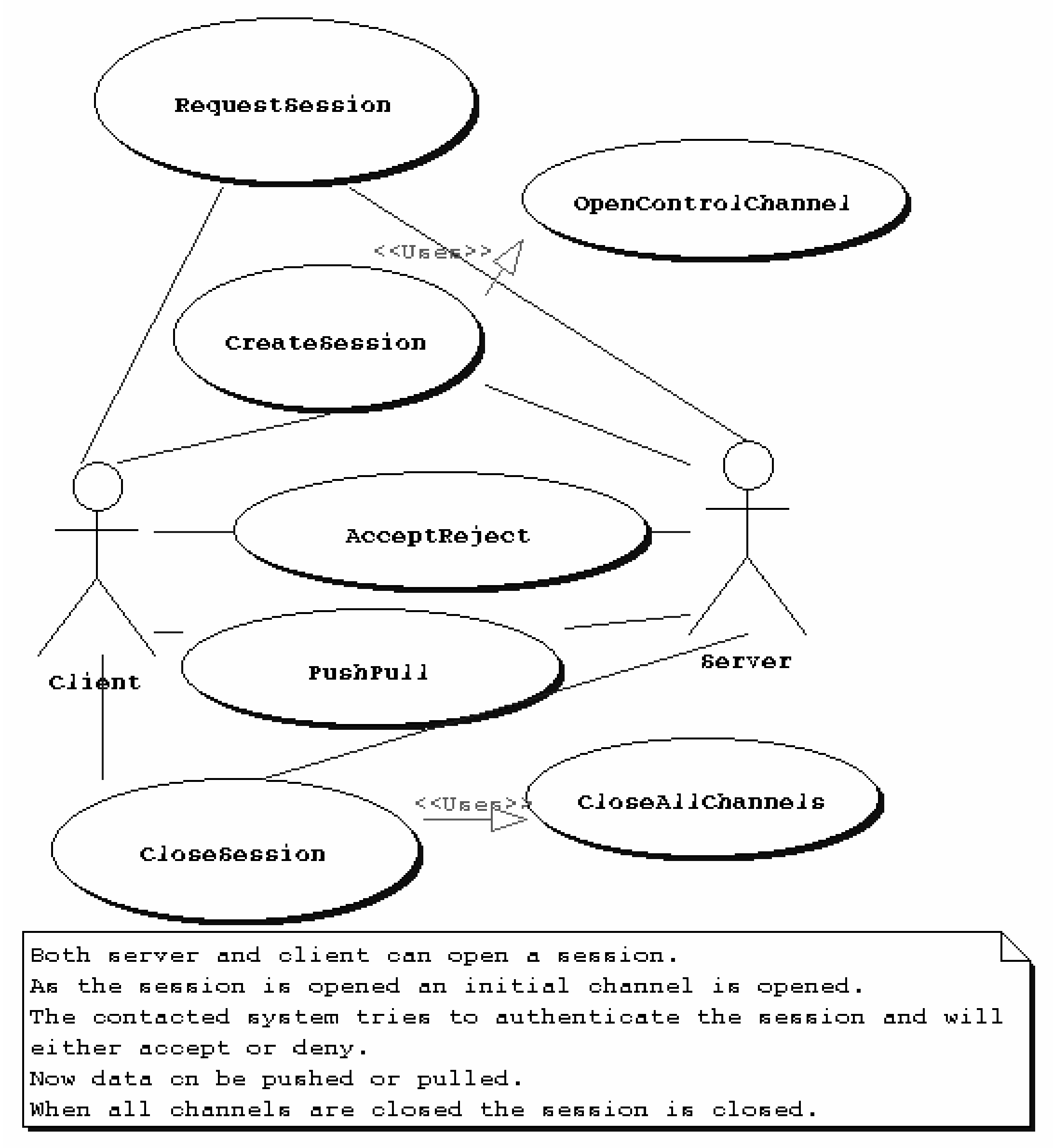
Push Use Case :



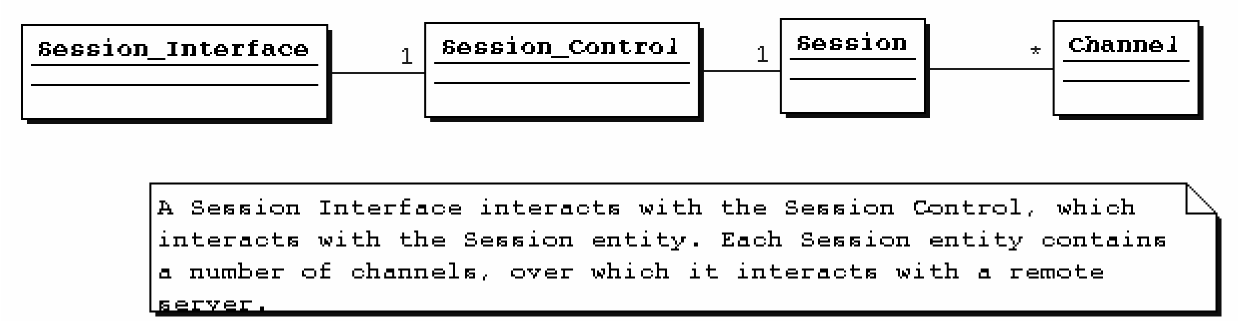
Pull Use Case :



Open / Close Use Case :

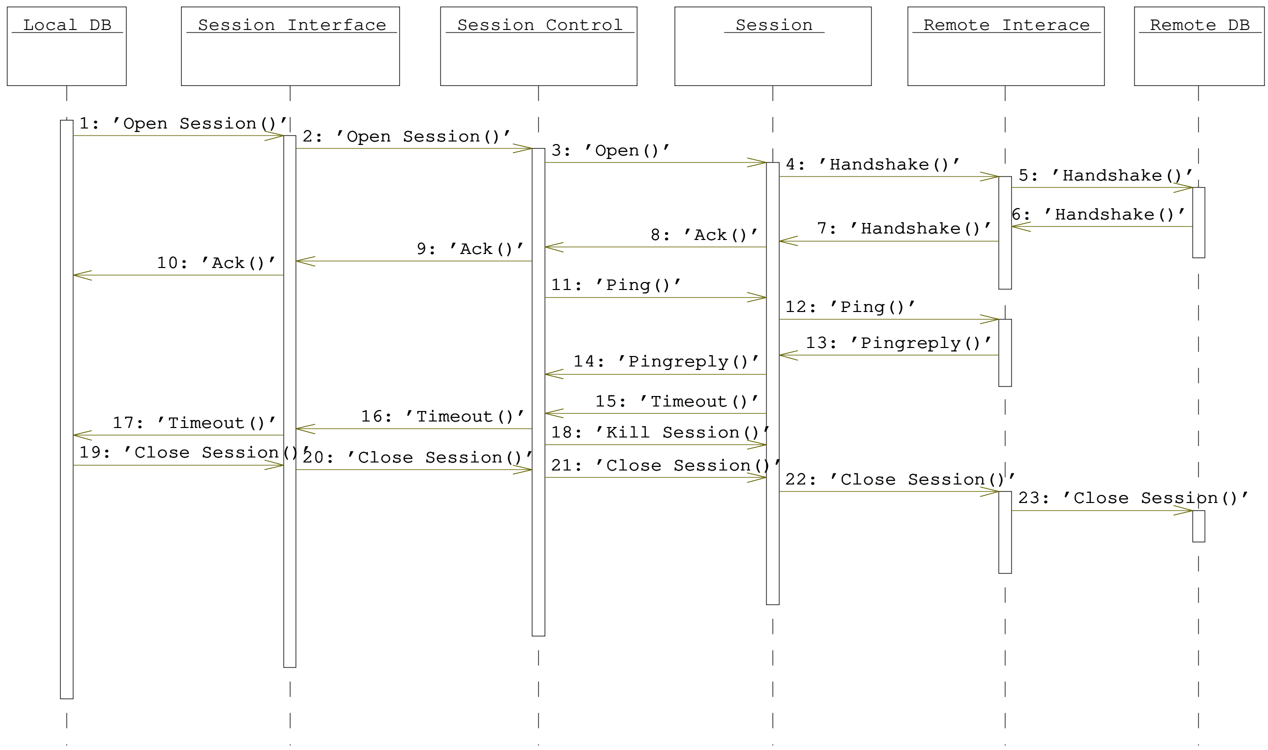


3.5.3 Object Model

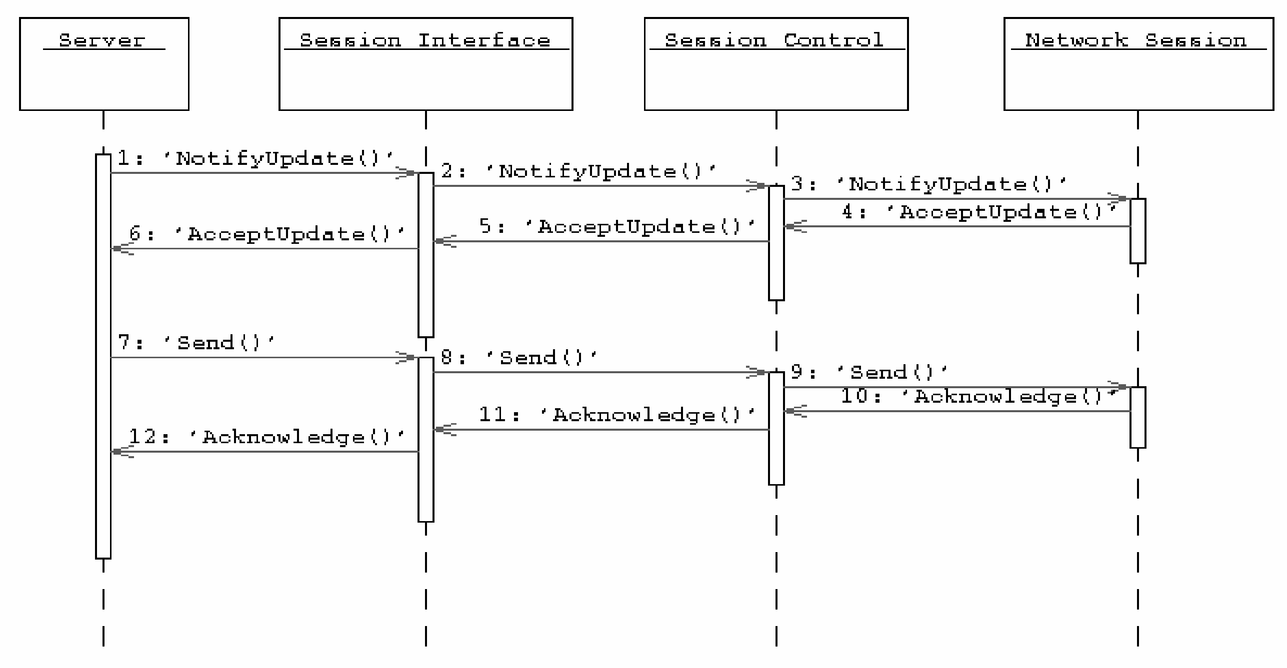


3.5.4 Dynamic Models

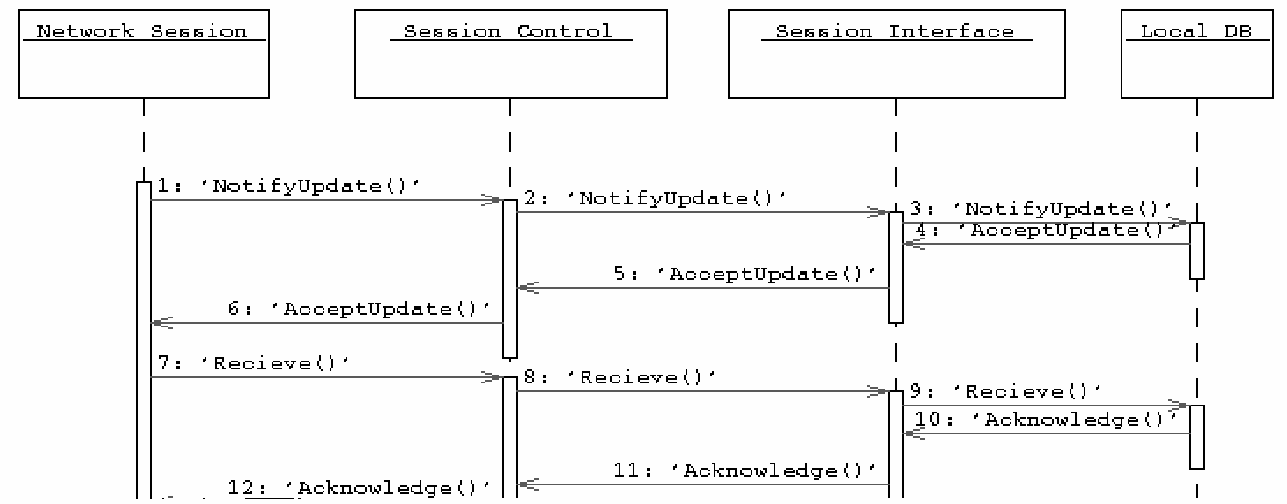
Open / Close Session :



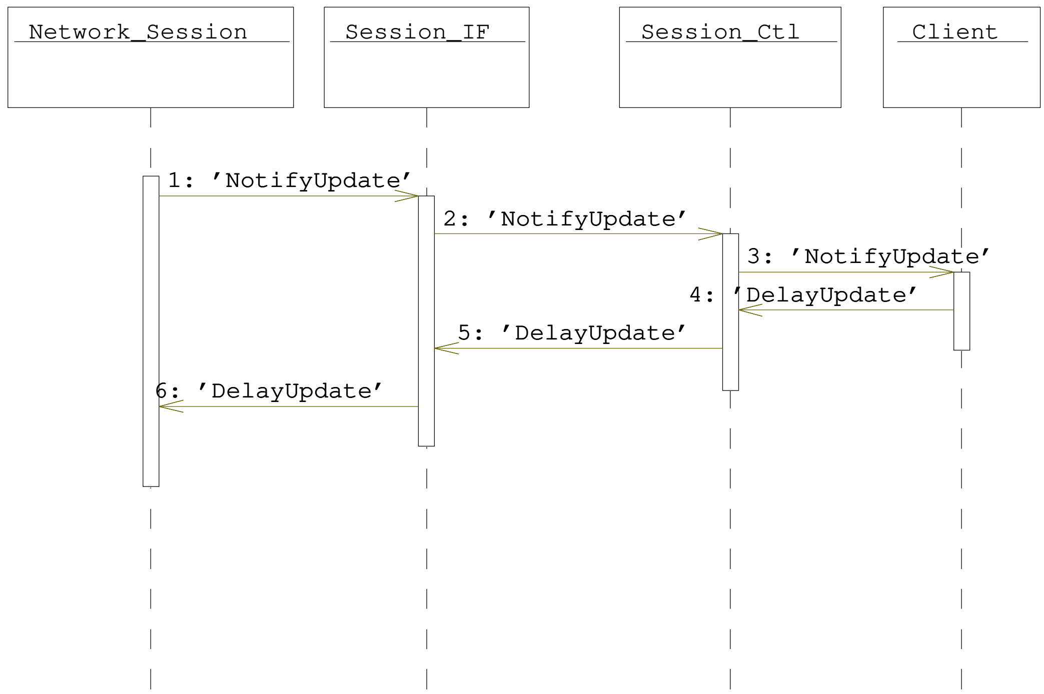
Server Side Push :

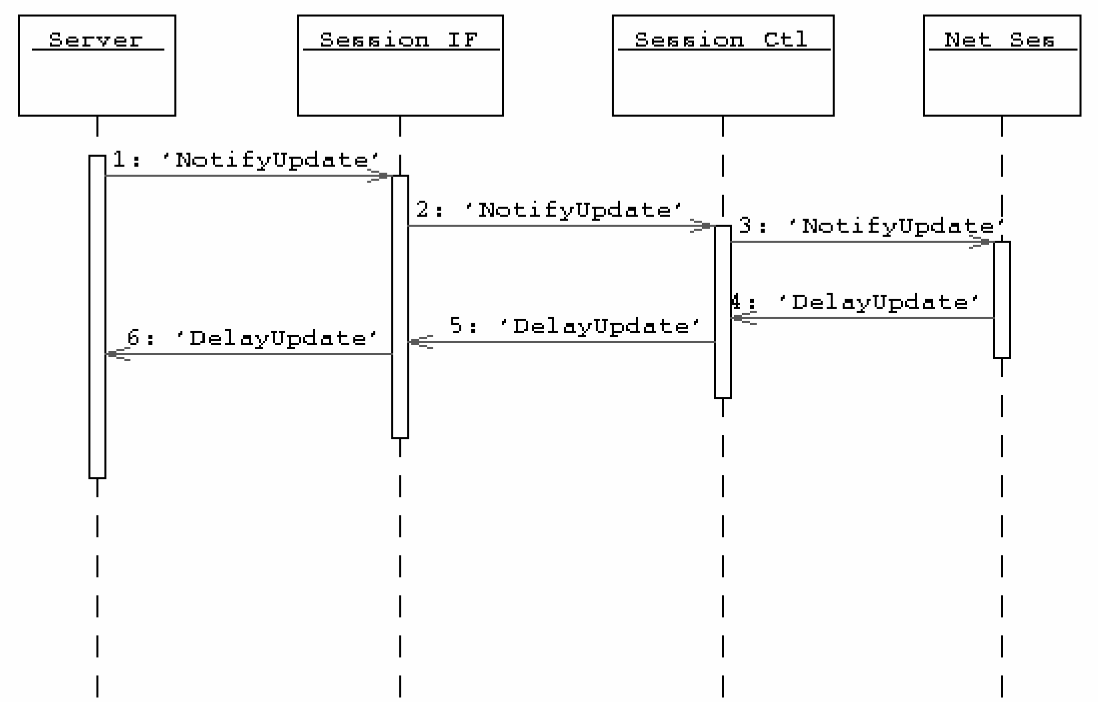


Client Side Push :

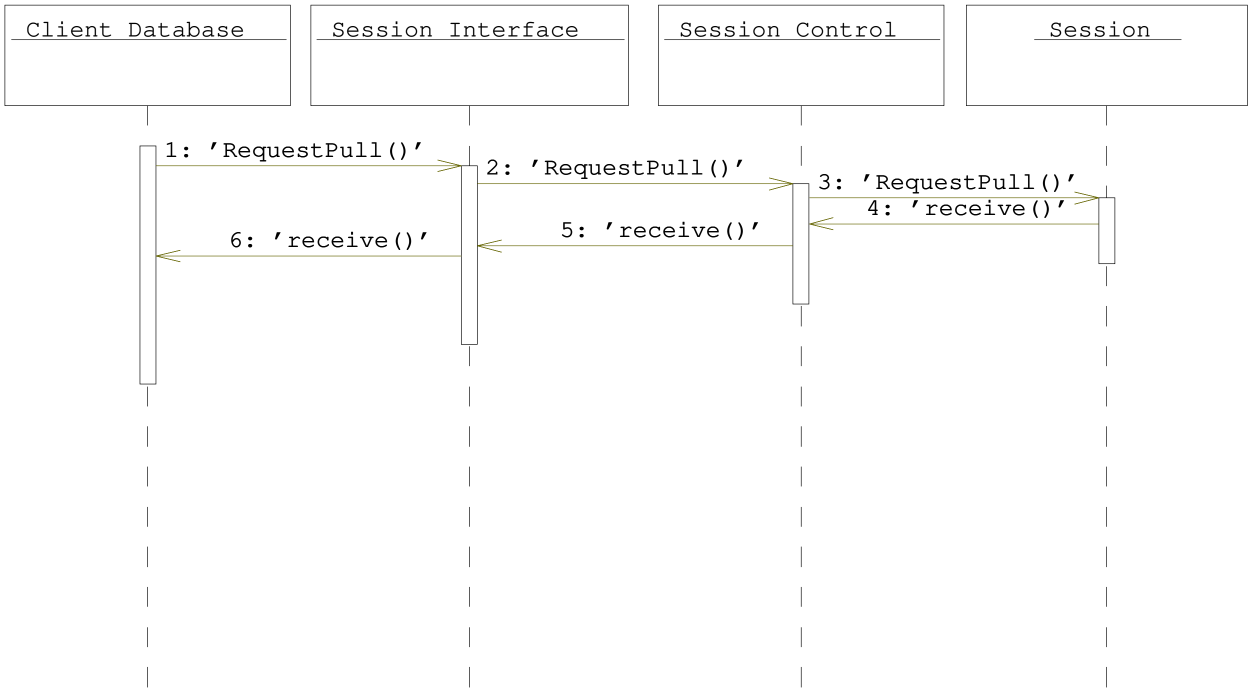


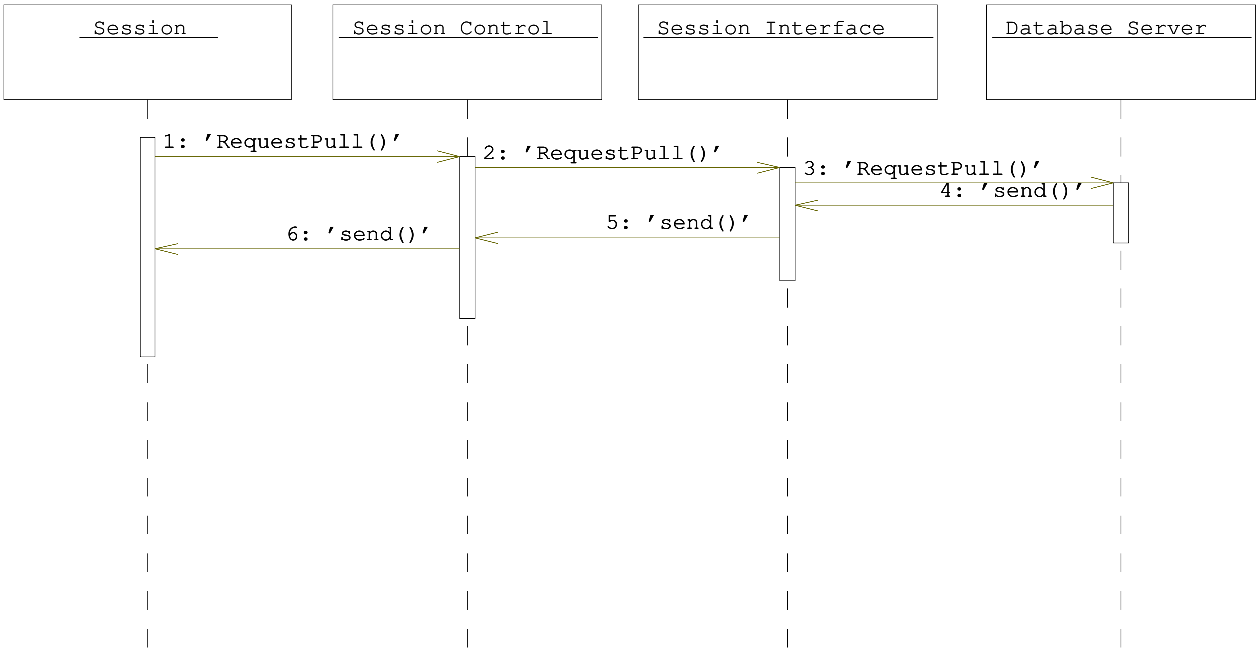
Push Delay (Client–Side / Server–Side) :





Pull (Client–Side / Server–Side):





3.5.5 User Interface – Navigational Paths and Screen Mockups

Network does not have any direct interaction with the users. Therefore, we do not have a user interface.