Call Center Customer Care System

(a case study)

Note: **this is a simplified and generalized description of a real system**

Introduction

The *Call Center Customer Care System* has been developed by Andersen Consulting for a large US telecommunication company. The primary function of the system is to support interactions with the customers that request new services (ex: new phone lines), changes in the configuration of the existing services (ex: phone number changes, long-distance company changes, or relocation), or report problems.

The phone company has over 19Mil customers. Considering how often, on average, a customer changes his/her service configuration, the system has to support up to 400 company representatives simultaneously at near 7X24 availability level. However, these representatives are not the only means by which a customer can request a change or report problems. For example, there exists a phone service (Quick Service) by which customers can communicate with the system. This is discussed in more detail later.

System Interactions

The C4 (Call Center Customer Care System) system interacts with a number of other systems, in particular with:
- A network provisioning system that makes physical changes to the network configurations and supports network management
- A Billing system
- A host of corporate DBs, and
- A number of downstream systems.

A high level structure of the system interactions is show in Fig. 1. Collectively, this is a big thing and rather complicated.
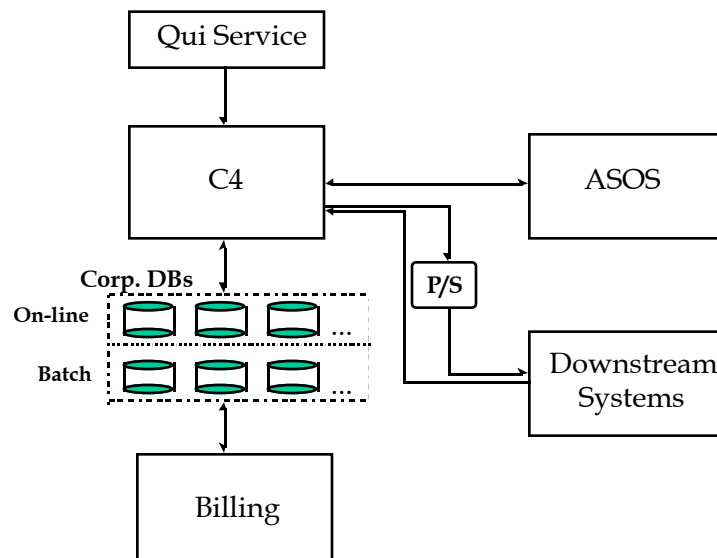
Interactions with NOSS

NOSS (Network Operations Support System) is the network management and provisioning system. It is being developed concurrently by a large 3rd party hardware/software company specializing in communication networks. Its functionality contains:

- Work force management - management of maintenance crews
- Provisioning - putting physical network components in place such as connections from the curb to the house
- Network creation - an peculiar name for maintaining information about new and existing physical network
- Activation - automatic turning on/off of services
- Network management that contains:
    - status monitoring and measurements
    - proactive maintenance
    - diagnostics, and
    - problem reporting, and
- Field access - a subsystem providing up-to-the-minute information for field technicians.

The NOSS architecture is based on the OSI CMIS (Common Management Information Service). The interface between C4 and NOSS is a large set of messages in the named-tag/value format. Each message inter-change is technically a synchronous pair of messages composed of a request and a reply. At the application level there are two types of interactions: 1) a request followed by a response that contains the requested information and/or confirmation of NOSS action, and 2) an overall interaction consisting of two message pairs; a) initial request followed by a reply containing a request identifier, b) an unsolicited message from NOSS containing the previously issued request identifier along with related data followed by a C4 reply acknowledging receipt of message.

The basic interactions between C4 and NOSS are as follows:

- C4 sends a Service Request (service order messages) to NOSS to perform service reconfigurations
- C4 send queries to NOSS about existing network status or capabilities. This is usually done while an agent is talking to a customer. For example, a request may be sent to NOSS about availability of service at a particular geographic address/service location or possible service activation dates, and
- C4 may request from NOSS to "lock" resources such as phone numbers for a service request.

Interactions with Downstream Systems

Downstream Systems are systems such as: Long-distance Carrier Services, 911 Service, Voice Mail, Interfaces to Phone Directory, and Revenue Collections (credit scoring and checking).

The C4 interacts with these systems in tow ways:

- It publishes requests via the Publish/Subscribe (P/S in Figure 1) component that the downstream systems should react to. For example, all new phone connections are published so the 911 service is connected to them in the required number of hours
- The downstream systems notify C4, via a direct asynchronous message, about significant business event that effect customer configuration. For example, the Long-distance Provider system may notify C4 that a client should be connected to a new long-distance company.

Note: one of the design challenges that we will discuss later was that direct requests from a customer may conflict with a similar request coming from an external system. For example, a customer may call to request a change of their long-distance carrier, say from Sprint to MCI, at the same time ATT sends notification that the same customer has now selected them as their long-distance carrier.

Interactions with corporate DBs and Billing

Some of the major functions of the billing system are: (1) invoice calculation for local services, (2) invoice printing for both local and long-distance services, (3) revenue reporting, and (4) bill inquiry and adjustment.

C4 does not interact with the Billing System, but it works against the same set of DBs. More specifically, C4 interacts with the on-line part of the corporate DBs, while the Billing Systems work with the batch copy of the on-line DBs (this is shown in Figure 1). Monthly billing is done in a number of cycles. Each cycle processes billing information of a set of customers. All updates to the data of the current cycle are halted (applied only to the on-line data) until the end of the cycle. The updates (at the end of the cycle) are done in batch and are transparent C4.

All customer data are stored in over 100 tables. C4 does not use all of them. During any customer conversation, C4 obtains customer basic profile from about dozen tables. Other tables are read and/or updated on demand.

C4

Functional description

C4 is an OLTP system that handles an interesting type of transactions. A transaction is initiated by a customer call, or so called Business Event. There are three types of events:
- Service Negotiations
- Account Management, and
- Trouble Call Management.

Each event is then divided into Tasks which are in turn broken into Activities. Tasks are groups of related activities that the representative and the customer have to complete. For example, if the negotiation is about the customer moving from one address to another, there will be a set of activities related to terminating some of the existing services, a set of activities related to obtaining the new address, and a set of activities related to negotiating new services and activation dates.

C4 must provide:
- Support for multiple concurrent tasks. For example, the customer should be able to negotiate a number of different services during the same call.
- Integrated support for completing activities (screen sequences, to-do list, context-sensitive data fields, etc.)
- Validation
    - of availability of requested service
    - completion of activities and tasks
    - integrity of customer data, and
    - integrity of the final requested configuration.
- Advice on available products and product "bundles"
- Resolution of conflicting events, and
- Support for interrupted and long-lasting conversations.

The last two points are particularly interesting. Resolution of conflicting event comes from multiple so-called Authors of events. For example, while a wife is negotiating a new phone line, the husband is using the phone company kiosk at is bank to request an ISDN line that will have an extra two phone lines. In general, events can come from:
- Direct conversations with company representatives (the case described here)
- Automated call center (the phone menu-type system)
- Kiosks (future), and
- Direct customer connections such as Internet (future).

Before C4 sends a service request to NOSS, it has to make sure that all related events have been combined or conflicts resolved.

The other requirement comes from the fact that a conversation with a customer can be interrupted (for technical reasons, for example) or suspended by the customer or the representative. The first case is rather obvious. An example of the second case when a customer that says something like "let me talk to my wife and I will call you back". In any case, C4 has to manage the context that persists and can be recalled.

Key architectural challenges

Here is a partial list of architectural challenges for C4. The challenges are not completely independent, so there is some repetition in the list. A number of challenges are implied by the execution architecture selected for the system. More about the architecture in the following section.

- Managing time and date effectively. Customer may want service changes in the future and this implies:
    - some form of a tickler system
    - ability to inform the customer about future changes of rates and/or services
- Interfacing with multiple authors of business events. As explained above, the main source of business events is direct conversation of an agent with a customer. However, other sources such as downstream systems, kiosks, etc. have to be accommodated. Important points:
    - business events from different authors may conflict with regards to the requested configuration at a service location
    - business events from different authors may be received and processed at the same point in time – business events from different authors become different service requests that must be cross validated to ensure a valid resulting configuration
- Architect something that is economically viable with a small set of customers and yet can grow to a very large network (i.e. 15 million customers)
    - it should not require high initial equipment investment
    - it should allow for "leaner" growth (in respect to cost(capacity) function)
    - it should be able to grow at a rapid rate (for example, 1000+ new customers a day)
    - identification, monitoring, and elimination of processing bottle-necks
- Validation of a requested service configuration should be done at near-real-time. This implies that:
    - C4 has to communicated with NOSS and other systems while guiding agents through tasks and activities, and
    - C4 may request to "lock" some of the network resources for a fixed amount of time (like a few phone numbers)
- Support for long-lasting, interrupted sessions. This issues has been described above
- Integrated (smart) performance support for company representatives
- Support a large number (e.g. 400+) of service representatives concurrently
    - a minimum of 100 service representative to start
- Near 7X24 application availability

Execution architecture

The system execution architecture is a standard three-tear C/S configuration as shown in Figure 2.

Win32 PC

Win32 PC

Win32 PC

WAN

HP9000

HP9000

HP9000

Billing DB

Enterprice DB

ASOS
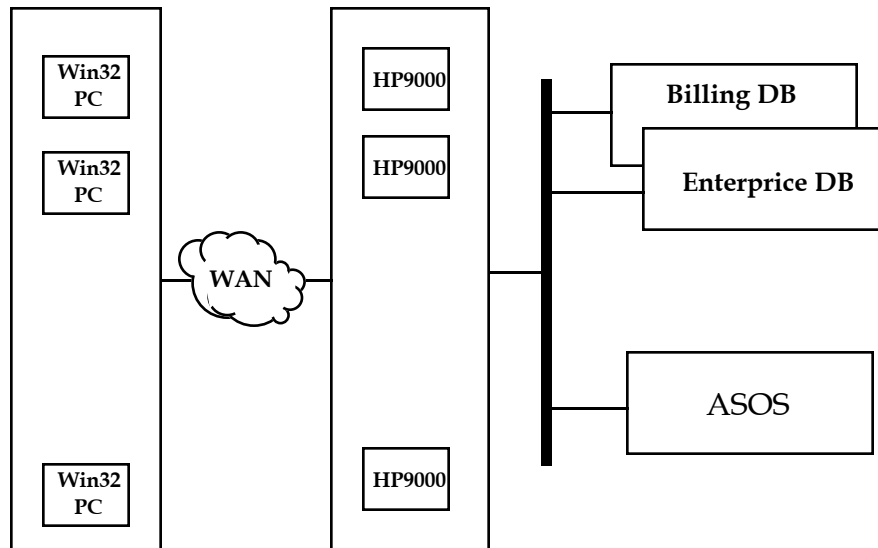
All agents workstations are PCs running the WinNT OSconnected to a LAN that is interconnect to a WAN. The middle layer is a cluster of HP9000 servers running UNIX and a TP Monitor that can load-balance between them. The TCP/IP communications protocol is used throughout the network. The back-end runs on a dedicated high throughput LAN. This LAN connects the Enterprise and Billing DBMSs to the servers as well as a TCP/IP connection to NOSS via a module that marshals massages between C4 and NOSS. C4 runs on the cluster of PC and HP9000.

Additional architectural requirements are as follows:
- No persistent data caching on the agent workstations to limit the implications of local failures
- No DBs at office locations
    - no administrators at local offices
    - no maintenance down-time, etc.
- The middle layer server cluster tuned for performance
    - possibility to add serves to increase throughput
- The back end tuned for DB performance
    - preferred place to do persistency
- Well engineered operations architecture
- High availability cannot be achieved by utilizing fault-tolerant hardware (this option is not economically viable)