



Universität-GH Paderborn

Fachbereich Wirtschaftswissenschaften

Diplomarbeit

**Personal Digital Assistants in Process-Driven Enterprise Sales Systems:
Concepts, Solutions, Business Value**

vorgelegt bei

Prof. Dr. Nastansky

im Juli 2000

vorgelegt von

Valentin Balzer

Weststraße 50

57392 Schmallenberg

Matrikelnummer 3369850

Studiengang Wirtschaftsinformatik

Acknowledgements

Without the support of the following people this work could not have been realised.

I would like to express my thanks to Prof. Dr. Ludwig Nastansky who was in charge of this thesis and supported me in the diploma procedure.

Dr. Anja Schliwka was the mentor of this thesis. I would like to thank her for her patience and for showing me how to focus the scientific aspects of the thesis.

Thanks go to the PAVONE AG, especially to Frank Iveson, for their efforts in acquiring and organising the project for the practical part of this work.

I would like to thank the SYMANTEC UK sales department, especially Heather Piggott, Cathrin Hackett and Phil Robbins, for their explanations about the real life requirements of a sales person.

Also I would like to thank all my friends who assisted me in any way, especially John Masters who brushed up my English knowledge.

Special thanks go to my parents who supported me during the whole time of my studies and stood by me in every situation.

Valentin Balzer
Paderborn, July 2000

Abstract

Personal Digital Assistants (PDAs) are used by sales people to store and access personal information when out of their offices. Sales information systems are designed to support sales people with all the information they need when dealing with their customers. Integrated workflow management features automate processes in the sales departments. The diploma thesis shows concepts and solutions for integrating PDAs into sales processes and how to get business value out of them.

Today's technology provides systems to distribute corporate information to PDAs and get field information back from them. The small size makes PDAs easy to manage anywhere and anytime but also restricts them in their data storage capacities and data input facilities. Integrating PDAs into corporate workflow environments means finding concepts and solutions which distribute a minimum of information to the right people at the right time, but also provide these people with all the corporate information they need for their work. It also means processing the returning information automatically without manual intervention. In a sales department integration of PDAs could speed up the sales processes and could help the sales person to keep up-to-date when on the road.

The diploma thesis deals with the use and the benefits of PDAs in sales departments. It shows the possibilities of integrating PDAs into the IT-infrastructure and especially into process-driven sales systems. It provides concepts for distributing corporate sales-specific information to such devices. A prototype for the integration of Palm OS-based PDAs into a PAVONE Espresso-based sales information system has been developed as part of the thesis.

Contents Overview

| | | |
|------|---------------------------------------|-----|
| I | Table of Contents | III |
| II | Content | |
| | 1 Introduction | 1 |
| | 2 Definition of Terms | 5 |
| | 3 Problem Analysis and Requirements | 15 |
| | 4 Technology Research | 21 |
| | 5 Concepts for the Integration | 37 |
| | 6 Integration Prototype | 47 |
| | 7 Conclusions and Future Perspectives | 61 |
| III | Bibliography | 67 |
| IV | Table of Figures | 71 |
| V | Table of Tables | 73 |
| VI | Table of Abbreviations | 75 |
| VII | Eidesstattliche Erklärung | 77 |
| VIII | Appendix | 79 |

I Table of Contents

| | | |
|----------|--|-----------|
| 1 | Introduction..... | 1 |
| 1.1 | Introduction to the Problem..... | 1 |
| 1.2 | Objectives..... | 2 |
| 1.3 | Structure of the Thesis..... | 3 |
| 2 | Definition of Terms..... | 5 |
| 2.1 | Personal Digital Assistant | 5 |
| 2.1.1 | History and Definition..... | 5 |
| 2.1.2 | PDA as Personal Information Manager | 6 |
| 2.2 | Workflow Management System..... | 7 |
| 2.2.1 | Basic Terms of Workflow-Management..... | 7 |
| 2.2.2 | The Workflow Reference Model of the Workflow Management Coalition | 9 |
| 2.2.3 | Using Workflow in the Office Environment..... | 11 |
| 2.3 | Enterprise Sales System | 12 |
| 2.3.1 | The Sales Process | 12 |
| 2.3.2 | Sales Force Automation Systems | 13 |
| 3 | Problem Analysis and Requirements..... | 15 |
| 3.1 | Sales Information Management for Sales People..... | 15 |
| 3.2 | Mobility Limitations of Notebooks..... | 16 |
| 3.3 | Administrative Considerations..... | 18 |
| 3.3.1 | Security Issues..... | 18 |
| 3.3.2 | Integration Issues..... | 18 |
| 4 | Technology Research..... | 21 |
| 4.1 | PDA Hardware | 21 |
| 4.1.1 | Palm OS..... | 21 |
| 4.1.2 | Windows CE..... | 23 |
| 4.1.3 | EPOC..... | 24 |
| 4.1.4 | Choosing the appropriate PDA..... | 25 |
| 4.2 | PDA Software..... | 27 |
| 4.2.1 | Native PDA Development Tools..... | 27 |
| 4.2.1.1 | Native Development Tools for Palm OS | 28 |
| 4.2.1.2 | Native Development Tools for Windows CE..... | 28 |

| | | |
|----------|--|-----------|
| 4.2.1.3 | Native Development Tools for EPOC..... | 29 |
| 4.2.2 | Rapid Application Development Tools..... | 29 |
| 4.2.2.1 | Pumatech Satellite Forms..... | 30 |
| 4.2.2.2 | GlobalWare Pylon Pro..... | 30 |
| 4.2.2.3 | SYWARE Visual CE..... | 31 |
| 4.2.2.4 | Feras Technologies CASL..... | 32 |
| 4.2.3 | PDA Databases..... | 32 |
| 4.2.4 | PDA Synchronisation Engines..... | 33 |
| 4.2.4.1 | Oracle iConnect..... | 33 |
| 4.2.4.2 | IBM Mobile-Connect and Extended Systems XTNDConnect..... | 33 |
| 4.2.5 | Architecture decision for the PDA Application..... | 35 |
| 5 | Concepts for the Integration..... | 37 |
| 5.1 | Handling the Dataload..... | 37 |
| 5.1.1 | Three-step Data Volume Reduction..... | 37 |
| 5.1.2 | Predefined and User-defined Data Selection..... | 39 |
| 5.1.3 | Business Rules for the Data Selection..... | 40 |
| 5.2 | Handling the Workload..... | 41 |
| 5.2.1 | Server-based Routing..... | 41 |
| 5.2.2 | Server-based Validation..... | 44 |
| 5.3 | PDA User Interfaces..... | 45 |
| 6 | Integration Prototype..... | 47 |
| 6.1 | The PAVONE Sales System..... | 47 |
| 6.2 | Tools and Architecture..... | 49 |
| 6.3 | The PDA Backend..... | 50 |
| 6.4 | The PDA Front-end..... | 51 |
| 6.5 | Enhancements of PAVONE Sales..... | 54 |
| 6.5.1 | Enhancements of existing Forms..... | 54 |
| 6.5.2 | User-defined Data Selection..... | 55 |
| 6.5.3 | Modifications to the Process Engine..... | 56 |
| 6.5.4 | Server-Based Routing and Server-Based Validation..... | 56 |
| 6.6 | Synchronisation Setup..... | 57 |
| 6.6.1 | Synchronisation..... | 57 |
| 6.6.2 | Data selection..... | 58 |
| 6.7 | Data Volumes on the PDA..... | 59 |

- 7 Conclusions and Future Perspectives61**
- 7.1 Conclusion.....61
- 7.2 Business Value62
- 7.3 Future Perspectives.....63

1 Introduction

The first chapter shall give an overview of the thesis. It starts with a short introduction to the problem and shows the overall objectives the thesis is aiming for. The structure of how these objectives have been reached are described in the last part.

1.1 Introduction to the Problem

Quite often a deciding factor in whether a company succeeds or not, aside from the quality of their products, is how the company cares about their current or potential customers. In many industries the quality of competing products is more or less the same, which has the effect that a customer decision on a specific product more and more depends on subjective factors.

This effect forces companies to adopt processes which are generally summarised by the term Customer Relationship Management (CRM). It combines all processes which are used both to canvass for new customers and to bind existing customers to the company. The sales function plays the decisive role, especially in the process of canvassing new customers. Information about a potential customer is usually gathered by the marketing department but the sales person is most often the first person with whom a customer comes in contact when dealing with a company. In any conversation with the customer, one of the main issues for the sales person is to have all information about the contact and the preceding conversations available, to convey a feeling to the customer of being taken seriously.

Nowadays a special area of CRM called Sales Force Automation (SFA) deals with methods for the process of a lead becoming a customer and provides systems which support sales people and sales management in their work. These are usually client/server-based systems with clients running on Windows-based operating systems. These systems take over the functions of collecting all information about the customer, providing this information for the sales person when facing the customer, reminding the sales person of customer-related events and tasks, and giving sales management an overview of the efficiency of their sales people.

Problems occur when sales people are not able to access the SFA system because they are in a situation where they cannot use their desktop PC or notebook. In these

situation a sales person still relies on paper-based information management or something similar which decreases productivity and can produce errors.

At this point Personal Digital Assistants (PDA) could help to bridge the gap where a PC is not usable. Due to their size they can be used nearly everywhere and anytime and are able to synchronise information with PC-based systems. At the moment PDAs are not integrated into corporate IT structures and the applications running on these devices focus on personal information management. Also the synchronisation capabilities are only usable for synchronising personal information to personal PC-based applications.

1.2 Objectives

The aim of this thesis is to develop concepts for the integration of PDAs into a process-driven Enterprise sales system as well as implementing the developed concepts as a prototype for an exemplary sales system.

The objective of the prototype is to demonstrate how to support the sales people in situations where they cannot rely on traditional PC technologies.

There are sales-related software solutions for PDAs on the market at the moment. These solutions are not only not integrated into the Enterprise sales systems, but also are developed simply as pure information systems without integration into the corporate sales processes.

The thesis will show different scenarios which demand that PDAs need to be integrated into the corporate IT-structures and be handled as a full part of this structure. Hereby consistent standards for security and administration are needed, just as they are set for any other part of the IT-structure.

For sales people on the road the benefits of integrating PDAs into Enterprise sales information systems is to have all information about their customers at their fingertips and to be able to use this information anytime and everywhere. This is an important aspect when aiming for increased customer satisfaction.

To get a high acceptance from sales people for such a system it is important to develop a solution which is easy to use for the sales person and decreases the time needed for getting the required information available on the road.

In summary, this means that the aim is to transform PDAs from a personal information manager into a corporate sales information manager fully integrated into the company's IT-structure.

1.3 Structure of the Thesis

First the basic terms and technologies of PDAs and process-driven sales information systems are introduced. Afterwards based on the current level of technology the problems sales people are facing when on the road are regarded. Then, after researching which technologies are available for an integration, the concepts needed for the integration are developed. In the end both the results from the technology research and the developed concepts are used to implement a prototype which integrates Palm OS-based PDAs into a process driven enterprise sales system based on the Espresso technology. In the end, after summarising the results, the business value of such an implementation is described and future perspectives are given.

2 Definition of Terms

This chapter defines the base terms of the three main parts of the thesis. The first part describes the basics about Personal Digital Assistants. The second part shows how Workflow Systems are built up and how process-driven aspects can give advantages for different kinds of systems. In the end Enterprise Sales Information Systems are described with regard to how they can benefit from process-driven aspects.

2.1 Personal Digital Assistant

After giving an overview about the history of Personal Digital Assistants this chapter describes the main use of Personal Digital Assistants in the area of Personal Information Management.

2.1.1 History and Definition

The term Personal Digital Assistant (PDA), also often called Handheld Computer, came up first when the UK technology company “PSION” introduced a device called “Psion 1” in 1983. With the size of a cigarette packet the battery-powered device combined the functions of a database, calendar, scientific calculator and a BASIC-like programming language with having a 16-character LCD display and a small alphanumeric keypad. It was the first device which provided a digital storage for appointments in conjunction with a pocket size. Also with PSION providing feature-enhanced follow-up models during the next years the next real milestone was set by the computer manufacturer “Apple” introducing the “Newton MessagePad” in 1993. Apple was the first manufacturer who used a touch-sensitive screen in combination with handwriting recognition software instead of a keyboard [cmp. PCTechGuide 2000]. Today’s PDA market is shared by several manufacturers as described in Chapter 4.1 in detail.

Nowadays PDA’s are designed to serve as a substitute for a common paper-based organiser. The digital battery-powered devices, which are still roughly of the size of their forefather “Psion 1”, act as a calendar, address book and notepad having a much larger storage capacity than a paper-based diary of that size. Additionally they provide features which cannot be handled by a paper-based organiser, e.g. electronic mail, and are able to connect to other computers, either through cables or wireless

networks and can often be enhanced by auxiliary devices to support their users in different situations [cmp. Encarta2000].

2.1.2 PDAs as Personal Information Manager

A Personal Information Manager (PIM) is a very fuzzily defined type of software. It can be described as a software application to be used for organising random bits of personal information [cmp. Webopedia 2000]. Most of the common software applications on the market combine applications for managing personal contacts, appointments and todo-lists with a simple notepad in which the main focus is on integrating any kind of digital information which can occur in this scope of application. In most cases the PIM's aim is to provide all functionality of a paper-based organiser while trying to simulate their look-and-feel. Also they have the advantage to be able to integrate into communication systems, e.g. email clients. To have the information also available when not having access to the computer, PIMs support their users with very sophisticated print-mechanisms to print the information directly in the format of the most common paper-based organisers.

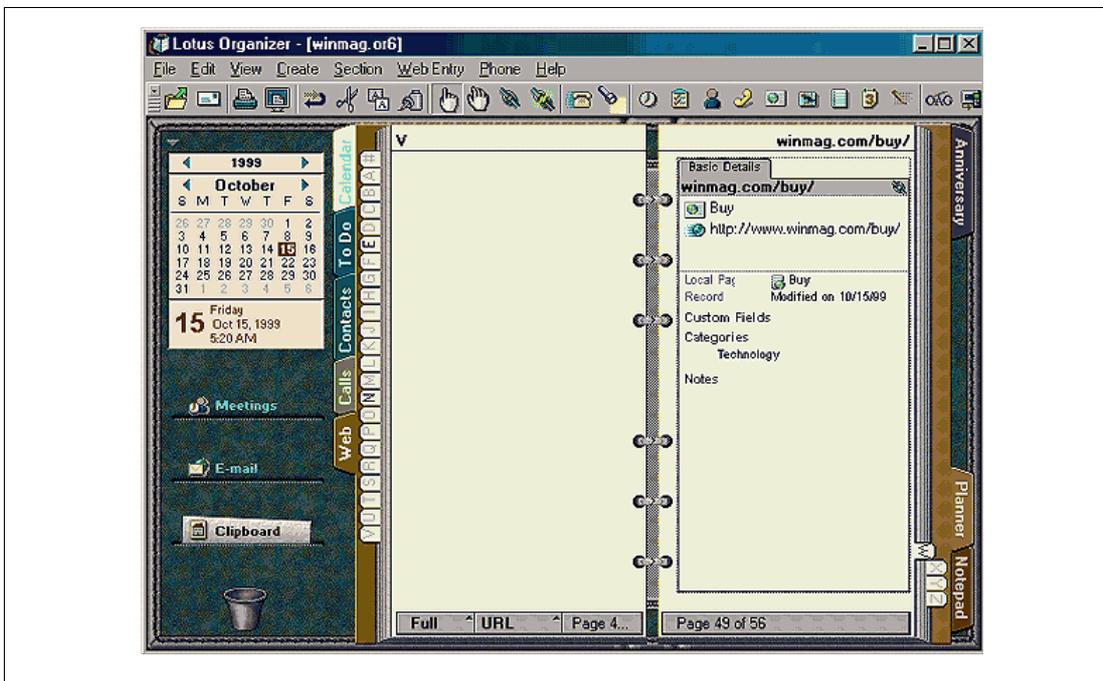


Figure 2.1: Screenshot of a typical PC-based PIM (Lotus Organizer 6.0)

Today's PDAs are usually shipped with a set of applications which provide nearly the same functionality as PIMs. Additionally they provide an interface to these PIMs so that the information of the PC-based PIM can be synchronised with the PDA-

based PIM. This means that one-way-synchronisation from the PC-based PIM to the paper-based organiser using printouts is replaced by a two-way-synchronisation between PC-based PIM and PDA-based PIM. The need for re-entering information into the PC-based PIM because it was changed on the paper-based organiser or the PDA can be reduced or even completely removed. Additionally the costs for the printouts and the time to find new or changed information in the PIM manually and printing it are removed when using PDAs, which increases the productivity of the user of the PIM [cmp. Nastansky et al. 1995, pp. 295]. Because of the use of synchronisation technology, PDAs can also be integrated into communication systems in the same way as PC-based PIMs. Regarding all these benefits a PDA can nowadays be seen as a real competitor for the paper-based organisers because PDAs provide much more storage capacity in a smaller size.

2.2 Workflow Management System

This chapter provides a description of the term of “Workflow Management System”. After defining the basic terms it explains the reference model of the Workflow Management Coalition. Then it describes the requirements for workflow management systems being used for business processes in the office environment regarding flexibility.

2.2.1 Basic Terms of Workflow-Management

For the better understanding of the following chapters it is reasonable to define some basic terms of the area of workflow management. To follow widespread standards, the definitions are taken from the Workflow Management Coalition (WfMC). The WfMC is a non-profit organisation, founded in August 1993, with the objective to define and distribute standards and terms for the use of workflow technology. The organisation unites manufacturers of workflow management systems and users of such systems as well as scientists working in this area.

Business Process (Synonym: Process)

“A set of one or more linked procedures or activities which collectively realise a business objective or policy goal, normally within the context of an organisational structure defining functional roles and relationships.” [WfMC 1999, p.10]

Workflow

“The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.” [WfMC 1999, p. 8]

Workflow Management System

“A system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications.” [WfMC 1999, p. 9]

Process Definition

“The representation of a business process in a form which supports automated manipulation, such as modelling, or enactment by a workflow management system. The process definition consists of a network of activities and their relationships, criteria to indicate the start and termination of the process, and information about the individual activities, such as participants, associated IT applications and data, etc.” [WfMC 1999, p. 11]

Instance (Synonyms: Workflow Instance, Process Instance)

“The representation of a single enactment of a process, or activity within a process, including its associated data. Each instance represents a separate thread of execution of the process or activity, which may be controlled independently and will have its own internal state and externally visible identity, which may be used as a handle, for example, to record or retrieve audit data relating to the individual enactment.” [WfMC 1999, p. 15]

Workflow Engine (Synonym: Process Engine)

“A software service or "engine" that provides the run time execution environment for a process instance.” [WfMC 1999, p. 57]

Workflow Enactment Service

“A software service that may consist of one or more workflow engines in order to create, manage and execute particular workflow instances. Applications may interface to this service via the workflow application programming interface (part of WAPI).” [WfMC 1999, p. 59]

2.2.2 The Workflow Reference Model of the Workflow Management Coalition

The WfMC has designed a reference model for standardising workflow systems. This chapter gives an overview about the basic aspects of the model which is shown in Figure 2.2 describing the main features of a workflow management system.

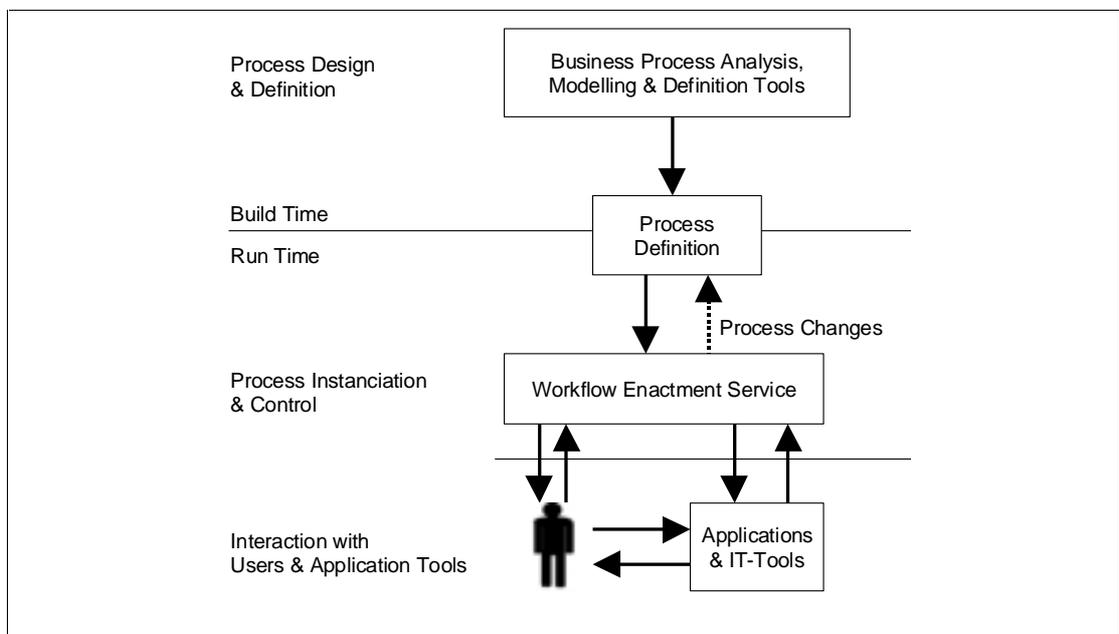


Figure 2.2: Workflow System Characteristics [Hollingsworth 1995]

Figure 2.2 divides a workflow management system into two stages. During the build time a real world business process is translated into a formal process definition using analysis and modelling techniques and appropriate tools [cmp. Hollingsworth 1995, p. 7].

During the run time process instances are initiated from the process definition. These instances contain information about a specific situation in the real world and are controlled by the workflow enactment service. The workflow enactment service hereby is responsible for assigning resources and for providing information for integrated or external applications [cmp. Hollingsworth 1995, p. 8].

Figure 2.2 only shows how the components of a workflow management system interact with each other. The reference model of the WfMC itself describes the architecture of a workflow management system. Figure 2.3 shows the main components and the interfaces between these components and the workflow enactment service.

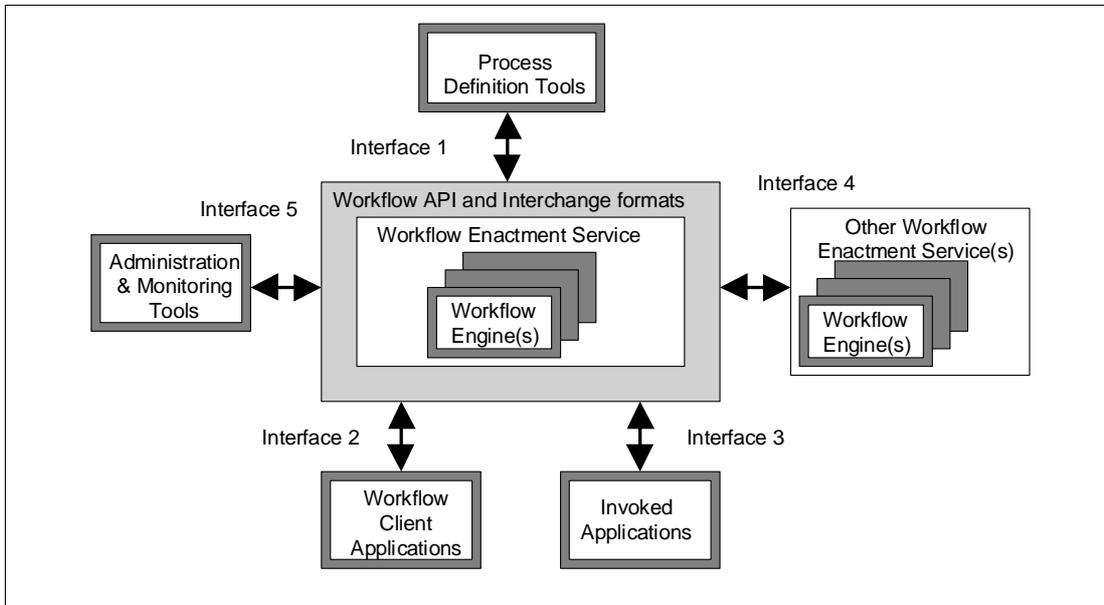


Figure 2.3: *Workflow Reference Model – Components & Interfaces* [Hollingsworth 1995, p.20]

The **Workflow Enactment Service** consists of a single or several workflow engines which provide the run-time environment for process instantiations and process activation [cmp. Hollingsworth 1995, p. 21]

The **Process Definition Tools** are used for analysing and modelling the business processes. Using a standardised interface makes it possible to combine a workflow enactment service with different process definition tools [cmp. Hollingsworth 1995, pp. 27-29].

The main component of the **Workflow Client Application** is the worklist handler providing the user an overview of his activities. This worklist is the start point for the user from which he calls all applications he needs to complete his activities. These activities cannot be automated because user input is needed [cmp. Hollingsworth 1995, pp. 31-35].

By contrast there are the **Invoked Applications** which can run fully automated. Because the invocation of any possible product in a heterogenous application environment is nearly impossible, a standardised interfaces is needed to transfer

information between the workflow enactment service and the invoked applications [cmp. Hollingsworth 1995, pp.35-37].

To integrate **Other Workflow Enactment Services** a standardised interface for the transfer of process activities is needed. This is a main aspect when regarding workflow management between different organisations [cmp. Hollingsworth 1995, p. 22].

The **Administration and Monitoring Tools** are used for supervising the process activities during the run-time. They are needed to gather statistical data which can be used for optimising the process definitions [Hollingsworth 1995, p. 44].

2.2.3 Using Workflow in the Office Environment

Workflow management systems as specified by the WfMC (cmp. Chapter 2.3.2) are mainly focussed on workflows having a predefined process definition. This kind of workflow has its origin in the production and logistics environment where such well-defined process chains are needed very often. In the office environment not every process can be defined and structured in that way, meaning that more flexibility is needed for the use of workflow [cmp. Nastansky / Hilpert 1994, p. 3].

It is possible to differentiate workflows by their grade of flexibility and their uniqueness. They can be categorised into three main groups:

- **Ad-hoc workflow** usually deals with unique and rather short-lived processes. Because of their uniqueness they are not worth predefining and this is also often not possible because the structure of the tasks is not known in advance. The complexity of such workflows varies because also the assignment of a single task is part of this workflow area [cmp. Nastansky / Hilpert 1994, p.4].
- **Semi-structured Workflows** are using well-structured, predefined process definitions, whereby parts of such a workflow can have characteristics of ad-hoc workflows. This could either be to handle exceptions but also to handle situations where parts of a workflow cannot be structured in advance [cmp. Nastansky / Hilpert 1994, pp. 5-9].
- **Standard Workflows** are completely structured and predefined. They can be fully supported by a workflow management system as defined by the WfMC.

Workflow management systems to be used in the office environment must be flexible enough for handling all these workflow types. On the one hand well-structured predefined workflows are needed, e.g. for holiday applications. On the

other hand many processes, e.g. customer requests, cannot be defined in advance [cmp. Nastansky / Hilpert 1994, p. 3]. The number of exceptions which can come up in this kind of business process is too large. This means trying to model all the process exceptions makes the process too unclear and confuses the user. Having the flexibility of an ad-hoc workflow such requests can be handled easily.

Another main consideration for workflow management in the office environment is to have a common information base which can handle any kind of information. The information which is collected within business processes are usually unstructured so that it is hard to match them into structured data patterns. Having a container which can store any kind of information, no matter whether well-structured or not, which is accessible within a process by all process members means being able to process any kind of request which can occur in the office environment [cmp. Nastansky / Hilpert 1994, pp. 2-3].

2.3 Enterprise Sales System

This chapter first gives an overview about the general sales process, about the flow of information and the organisational units involved in it. Afterwards it describes what kinds of systems are needed to form a fully integrated Enterprise Sales System.

2.3.1 The Sales Process

The process of selling, also called the sales cycle, is a very sophisticated process of the business area. The sales department has to work together closely with the marketing department and customer service. An example sales process as shown in Figure 2.4 starts at the marketing department with incoming leads given to the sales department. From there the lead goes through the several stages from being analysed to, in the best case, placing an order. At that point the lead is not owned by the sales department any more but is taken over by the customer service department. The cycle closes with information about the lead given back to the marketing department which needs the information for creating new campaigns for existing customers. During the whole process on the one hand information is developed and categorised into process information and information about lost and won sales. On the other hand the sales people are supported by information from former sales processes to optimise the current process.

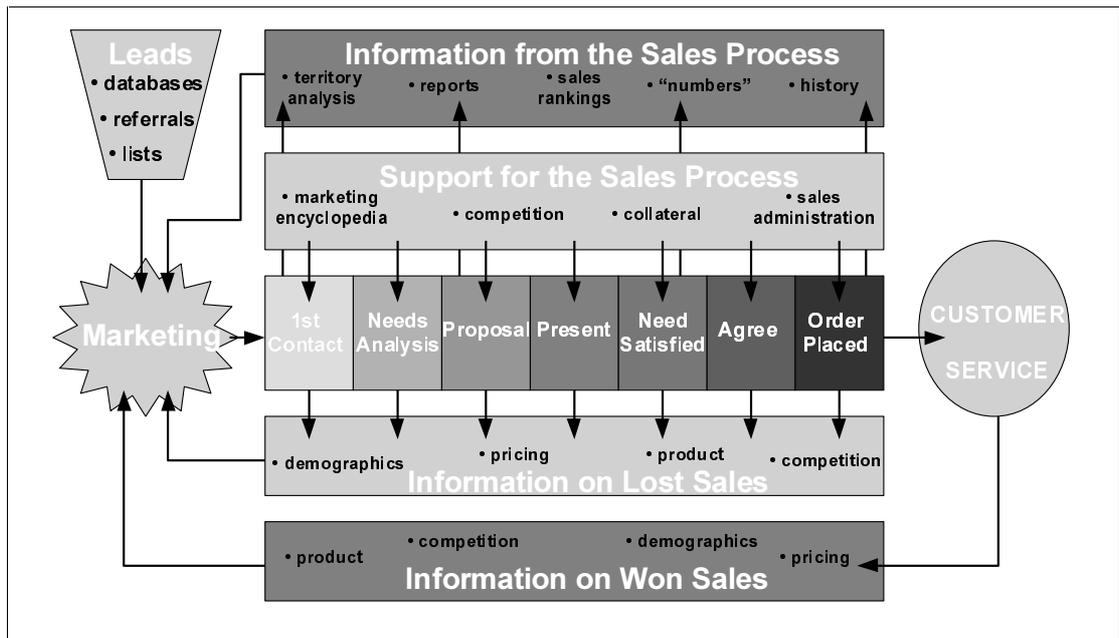


Figure 2.4: The Sales Process [McCrea 1999]

2.3.2 Sales Force Automation Systems

Sales Force Automation (SFA) describes systems which should support the sales team during the sales process described in Chapter 2.3.1. The process from the first contact to a placed order can be developed using workflow management systems as described in Chapter 2.2. The process as shown in Figure 2.4 only gives an overview of the process. In reality the single steps from the first contact with a lead to a successful sale are divided into many single tasks which are not known in advance, e.g. it could happen that not only one, but several proposals are sent out. This means that the flexibility considerations of the workflow management system as described in Chapter 2.2.3 are of a high importance when such a system is used in the sales cycle. The workflow management system must be linked with a contact management system to hold all the information about the lead plus the information which is developed during the sales process. This contact management system also must support customer service, when being in contact with the customer after the sale, and the marketing department which needs the information from former sales processes to create their marketing campaigns.

Usually SFA is seen as part of customer relation management (CRM) [cmp. Shiang / Hodges 1999], whereby SFA handles the process of a contact from being a potential customer to becoming an existing customer.

The only aspect missing in this process is the integration of the corporate financial systems which handle the invoicing and other financial tasks based on the information from the successful sales. When talking about Enterprise Sales Systems this aspect must be regarded. This means that at a specific point in the process the information about sales must be transferred to the corporate financial systems which are usually transaction-based systems made for storing large volumes of highly structured data. At first sight this seems to be only an additional interface to another type of system. But the transfer must be considered with highest accuracy because the transactional systems often are the calculation basis for taxes. This means that the sales information system needs additional security mechanisms to handle the transfer and changes of already transferred information.

3 Problem Analysis and Requirements

This chapter gives an overview about the problems which occur because of the increasing sales people mobility. Chapter 3.1 shows the current situation and how sales people try to adapt to changing requirements. Chapter 3.2 describes the requirements to current technologies regarding aspects of mobility and Chapter 3.3 shows the demands to mobile technologies regarding administrative and security issues.

3.1 Sales Information Management for Sales People

Recently working habits of sales people have changed because they had to adapt to the fact that products from different suppliers have become interchangeable. In some markets neither the features of a product nor its price nor its quality is a deciding factor for the customer. He is more and more interested in the service offerings and how the supplier looks after his customers. This means for the sales person that one of his main requirements is to have all customer information available whenever he gets in contact with the customer. Sales information systems as described in Chapter 2 focus on this aspect, help the sales people handling their sales contacts and forecasts, and support them in organising their tasks. But usually a sales information system needs to access Enterprise databases meaning that the device running the sales information system must be network connected. The performance of today's notebooks, combined with powerful replication technologies, enable a sales person to use a sales information system on his notebook without being connected to a network, e.g. at the customer's office. But nowadays sales-related conversations do not only take place within offices. Some examples of situations where sales-related information is needed could be: at business lunches or dinners, during leisure activities, on flights or between flights on airports. In these situations, for different reasons which are described in Chapter 3.2, a sales person cannot use his notebook to have specific information, e.g. contact details, correspondence history, schedule or task lists, available. This means they have to use autonomous devices like paper-based organisers or PDAs which can synchronise data with their PIM (cmp. 2.1.2). A paper-based organiser is usually used in such a way that the sales information is printed on a format fitting into the organiser. The sales person works with the printout and manually types the changes into the sales information system when back

in his office. When using a PDA he has to transfer the desired information to his PIM which synchronises with his PDA. When back at the office, all changes are synchronised back to his PIM and must be transferred to the sales information system manually.

No matter which method is used, new or changed information has to be re-entered into the Enterprise sales information system. It both takes time and thereby costs money but also it poses a serious source of possible errors.

3.2 Mobility Limitations of Notebooks

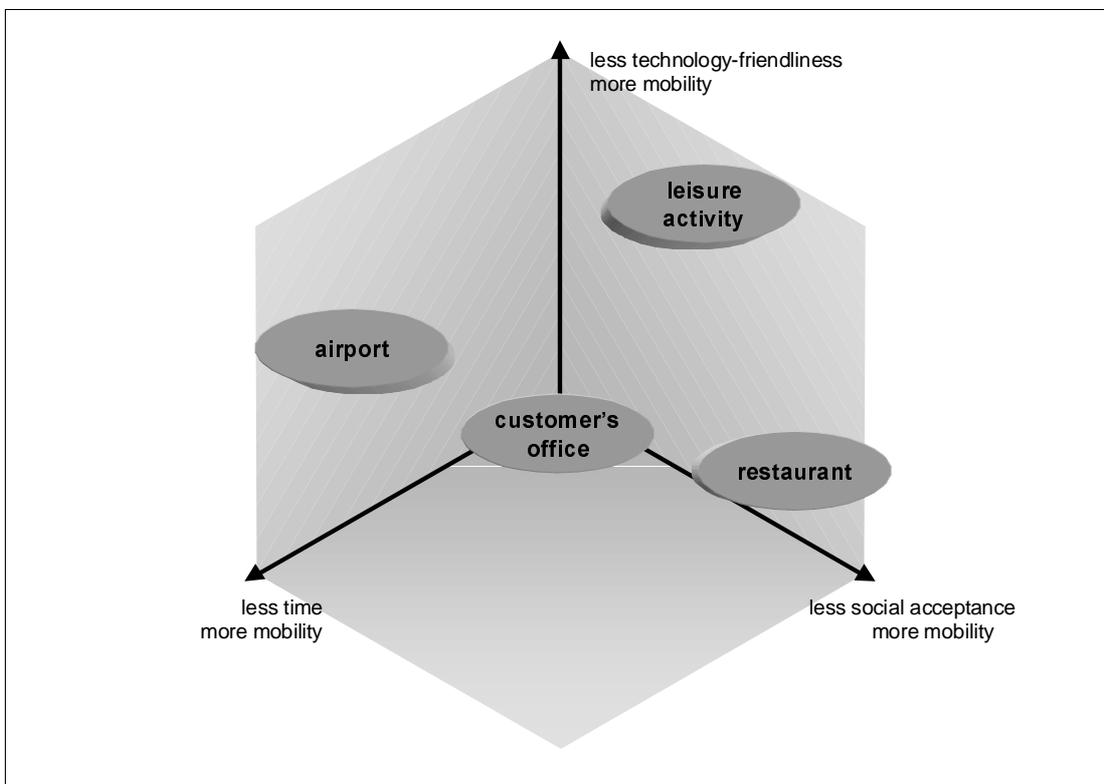


Figure 3.1: Mobility Aspects of some exemplary Places

As mentioned in Chapter 3.1 notebooks cannot be used in any situation. Most of these situations can be categorised in the following way:

- Time
The time for a sales conversation can differ from several hours to a few minutes.
- Technology-unfriendly environment
A power supply for high voltage devices it not always available and a secure base to site such a device is often missing. Also aspects like humidity, dirt or lighting conditions can make the use of such devices more difficult.

- Social acceptance of technology

The visible use of technology devices is not always welcome by other people. This arises mostly in areas people use for leisure or relaxation, and not for business activities.

Any place or situation can be categorised in this way. The result is a level of mobility which is needed for the device used in this place or situation. Figure 3.1 shows examples of how places can be categorised in this way.

The notebooks which are on the market at the moment are deficient in some of these aspects in the following ways:

- Battery life

The battery of a high performance notebook lasts for a few hours only. So the user relies on a power supply or an additional battery when he wants to use the system for the whole day.

- Size & weight

The smallest notebooks which have the performance to run a sales information system still have such dimensions and weight that they need to be placed on a desk or something similar to be used effectively.

- User interface

The common Windows-based user interfaces are designed to be used with two hands. This design increases the need for a robust base on which to place the notebook.

- Start-up time

A notebook equipped with a Windows-based operating system takes nearly a minute to start, plus the time to initialise the sales information system. In lots of situations where only a small piece of information is needed this is not suitable. Power save or sleep mode technologies can speed up the start-up time but this will reduce the battery life.

| | Manufacturer | Model | Dimensions in mm (H x W x D) | Weight in kg | Battery lifetime in hrs* |
|------------------------|--------------|-----------------|---------------------------------|--------------|--------------------------|
| Standard- Notebooks | IBM | Thinkpad 600 | 36 x 300 x 239 | 2.13 | 3.0 |
| | Sony | Vaio PCG-Z505HS | 44 x 307 x 261 | 2.66 | 2.5 |
| Mini- Notebooks | IBM | Thinkpad 240 | 27 x 260 x 202 | 1.35 | 1.8 |
| | Sony | Vaio PCG-C1XD | 29 x 248 x 153 | 1.00 | 1.5 |

*according to the manufacturer

Table 3.1: Notebook Dimensions and Battery Life of sample Manufacturers

These reasons cannot be seen as discreet. Some of them are interdependent, e.g. there is a trade-off between notebook dimensions and weight and battery lifetime as shown in Table 3.1.

3.3 Administrative Considerations

As mentioned in Chapter 3.1 sales people use different devices, either paper-based or digital, for their information management in situations where they cannot rely on a notebook. From an administrative perspective this brings up some issues to be considered when building a strategy for the use of mobile devices in an Enterprise.

3.3.1 Security Issues

No matter whether the sales people are using small paper-based organisers or PDAs, these devices are easily stolen by pickpockets because of their size, not only because of their intrinsic value but also because of the stored information [cmp. Kennedy 1999]. Also a small device is more likely to be lost by its owner than a larger one. Nowadays notebooks can be secured so that the stored information cannot be used by anyone other than its owner. The security levels range from using passwords for the use of the machine up to strong encryption of all data stored on the harddisk of the device. In this aspect paper-based organisers fail completely. There is no way to secure the information stored in the organiser so that it cannot be used by anyone else than its owner.

Depending on the model, PDAs can be secured up to a specific level, but the standard security features provided are usually not very strong and can be by-passed by specialists.

Another security issue arising from the use of PDAs is the possibility to use a modem to dial into corporate networks whereby the login information like passwords can be stored on the device itself [cmp. Soane 1999]. In the worst case this means that somebody who finds the PDA could get access to corporate databases and gets the ability to act in the name of the owner of the device, e.g. for sending emails.

3.3.2 Integration Issues

When thinking about integrating PDAs into Enterprise systems of any kind there is one new aspect which has to be considered. Formerly there was the chance to define

hardware and operating system standards for a device to be integrated into the IT-environment. The major concern when integrating PDAs is that they mainly have been purchased privately by the sales people as a substitute for their paper-based personal organisers already. The aspects of the sales person's buying decision are usually different from the criteria of someone looking after a company's IT strategy. A sales person chooses a specific device based on aspects like price, design or recommendations by colleagues, whereas the IT strategy focuses on TCO, security and integration aspects. The consequence of the private purchases is a multiplicity of PDAs of different manufacturers and a variety of software packages running on these devices.

As a matter-of-fact the number of support queries the IT departments have to deal with increases, although the PDAs are not a property of the company and not supported. These queries mainly do not concern the PDA itself but the connection between PDA and computer, e.g. how to use the infrared interface.

When starting to integrate PDAs into enterprise systems this diversity needs to be considered, because it will be difficult to force the users to a specific device if they have already purchased a different one. Nevertheless a policy is needed for the future purchases of PDAs to reduce the number of different PDAs and the consequent support queries.

4 Technology Research

This chapter gives an overview of the current PDA technologies. First it shows the features and differences between selected PDAs based on the three most popular operating systems on the market. Then it looks at the application development tools and the possibilities of synchronisation with Enterprise databases and groupware platforms.

4.1 PDA Hardware

As mentioned in Chapter 3, there is a variety of different PDAs available on the market, which can be categorised by their operating system. There are three main operating systems which are used by most of the manufacturers, but there are still some manufacturers which use their own proprietary operating system for their devices. This thesis will focus on the three main operating systems only, which are Palm OS, Windows CE and EPOC, because they support several industry standards, and because a wide range of additional software and hardware is available for these devices. Example devices of each operating system are shown in Figure 4.1.



Figure 4.1: Example PDA devices of the dominant PDA operating systems

4.1.1 Palm OS

Palm OS is an operating system made by Palm Inc. which is the market-leader for PDAs at the moment. Palm OS is used mainly for Palm's own product range of PDAs and licensed to some other manufacturers like Handspring or TRG Products. There are discussions to use it as an operating system for mobile phones enhanced

with PDA features, but only prototypes exist at the moment [cmp. Craig / Leyden 1999].

Palm OS devices are pen-based and provide a special text entry system, called “Graffiti”, which forces the user to write the characters in a special untypical way, but it is easy to learn and has a very low error rate for the character recognition. Additionally a foldable full-size clip-on keyboard is available for most devices using Palm OS.

| | Palm IIIxe ¹ | Palm IIIc ¹ | TRG Pro ² | Palm V / Vx ¹ | Handspring Visor / Visor Deluxe ³ |
|--------------------------------------|---|---|---|---|--|
| Size (H x W x D) in cm | 11.9 x 8.1 x 1.8 | 12.9 x 8.1 x 1.7 | 11.9 x 8.1 x 2.0 | 11.4 x 7.9 x 1.0 | 12.2 x 7.6 x 1.8 |
| Weight in g | 170 | 193 | 170 | 113 | 153 |
| Palm OS Version | 3.5 / 3.3 | 3.5 | 3.3 | 3.3 / 3.5 | 3.1 (not upgradable) |
| Screen | FSTN LCD 16 grey scale | Active TFT 256 colours | FSTN LCD 16 grey scale | FSTN LCD 16 grey scale | LCD 16 grey scale |
| Memory (RAM) | 8Mb | 8 Mb | 8 Mb | 2 Mb / 8 Mb | 2 Mb / 8 Mb |
| Battery Type | 2 AAA batteries | Rechargeable Lithium-ion | 2 AAA batteries | Rechargeable Lithium-ion | 2 AAA batteries |
| Battery Lifetime ⁴ | 8 weeks | 2 weeks | 8 weeks | 4 weeks | 8 weeks |
| Ports | RS232 serial port IrDA infrared port |
| Expansion Slot | - | - | CompactFlash type I/II card slot | - | Springboard module port |
| Price in US\$ | 249 | 449 | 329 | 329 / 399 | 179 / 249 |
| Comments | | | Enhanced Audio system | | Built-in Microphone |

¹ [cmp. Palm 2000]

² [cmp. TRG 2000]

³ [cmp. Handspring 2000]

⁴ based on typical use profile according to the manufacturer [cmp. Handspring 2000, Palm 2000, TRG 2000]

Table 4.1: Specifications of sample Palm OS-based PDAs

All current devices are equipped with a screen of 160x160 pixels to be used with or without backlight. The devices are powered by either two AAA-batteries or a rechargeable built-in lithium-ion battery and can connect to a PC either by using a docking station or an infrared port. For exchanging information between PC and PDA a technology called “HotSync” is used. The Hotsync engine consists of several

modules called “Conduits” which transfer the information from a single application on the PDA to an application on the PC. These Conduits can be used either for the “Palm Desktop” shipped with the PDA, or in addition for most of the PIMs on the market. In this way it is the only technology on the market which also serves PCs running on other operating systems than Windows, e.g. MacOS. All Palm OS devices ship at least with a software package containing Date Book, Address Book, ToDo List, MemoPad, Calculator, Expenses, Mail and Security.

4.1.2 Windows CE

| | Casio Cassiopeia E-100 / E 115 ¹ | Philips Nino 210 ² | Philips Nino 510 ² | Compaq Aero 1500 ³ | Compaq Aero 2100 ³ |
|--------------------------------------|--|---|---|---|---|
| Size (H x W x D) in cm | 13.0 x 8.3 x 2.0 | 13.7 x 8.4 x 2.0 | 13.7 x 8.6 x 2.3 | 13.0 x 7.9 x 1.3 | 13.4 x 8.5 x 2.0 |
| Weight in g | 255 | 227 | 244 | 143 | 260 |
| Windows CE Version | 2.11 | 2.11 | 2.11 | 2.1 | 2.1 |
| Screen | 240 x 320 TFT 65,535 colours | 240 x 320 LCD 4 grey scale | 240 x 320 TFT 256 colours | 240 x 320 STN 16 grey scale | 240 x 320 TFT 256 colours |
| Memory (RAM) | 16 Mb / 32 Mb | 8 Mb | 16 Mb | 16 Mb | 24 Mb |
| Battery Type | Rechargable Lithium-ion | 2 AA | 2 AA | Rechargable Lithium-ion | Rechargable Lithium-ion |
| Battery Lifetime ⁴ | 6 hours | 12 hours | 8 hours | 14 hours | 10 hours |
| Ports | RS232 serial port IrDA infrared port | RS232 serial port IrDA infrared port | RS232 serial port IrDA infrared port | RS232 serial port IrDA infrared port | RS232 serial port IrDA infrared port |
| Expansion Slot | CompactFlash type I/II card slot | CompactFlash type I/II card slot | CompactFlash type I/II card slot | CompactFlash type I card slot | CompactFlash type I card slot |
| Price in US\$ | 499 / 599 | 299 | 429 | 299 | 399 |
| Comments | Additional Lithium-ion backup battery | | | | |

¹ [cmp. Casio 2000]

² [cmp. Philips 2000]

³ [cmp. Compaq 2000]

⁴ continuous use according to the manufacturer [cmp. Casio 2000, Compaq 2000, Philips 2000]

Table 4.2: Specifications of sample Windows CE-based PDAs

Windows CE is an operating system developed by the Microsoft Corporation. It focuses not only on PDAs, but also on any kind of embedded systems e.g. TV-Set-Top-Boxes or fuel stations [cmp. Murphy 1999]. The versions of Windows CE used on PDAs have nearly the same look-and-feel as Microsoft’s PC operating systems,

e.g. Windows 98. On the one hand it is a benefit that the user interface is well known to most of the users, on the other hand it is not especially designed for handheld devices and lacks in some points which are described in Chapter 5.3.

Several hardware manufacturers like Casio, Compaq or Philips develop PDAs running on Windows CE. The majority of the devices is pen-based and uses handwriting recognition technologies. There are also keyboard-equipped devices on the market, but regarding their size and weight they are closer to the range of mini-notebooks than to the PDAs. Additionally all Windows CE devices are provided with sound systems and microphones so that they can be used as a digital sound recorder as well. The devices are mainly powered by rechargeable lithium-ion batteries and use serial or infrared ports to connect to PCs or other PDA devices. The manufacturers ship their devices with different software packages which contain at the minimum an address book, calendar, notepad, sound recorder and email client. The applications synchronise with the Microsoft Office package, including Microsoft Outlook as PIM, on the PC but can be adapted to other PIMs as well.

4.1.3 EPOC

EPOC is an operating system developed by PSION PLC. The devices equipped with EPOC are mainly manufactured by PSION itself, but nowadays they also license their technology to mobile phone manufacturers, e.g. Ericsson. The devices are equipped with keyboards for text entry and have touch-sensitive screens to be used with a pen. Because of their integrated sound systems and microphones they can be used as sound recorders as well. Their screens are made in a landscape format that allows a better text editing than the upright formats of Windows CE and Palm OS. They connect to a PC using a serial or infrared port. Software packages include all PIM applications which synchronise with all major PC-based PIMs plus a word processor and spreadsheet application to synchronise with all major Windows-based word processors or spreadsheet applications.

| | PSION Revo ¹ | PSION 5 Series ¹ | PSION 5mx Series ¹ | Ericsson MC 218 ² |
|-------------------------------|--|---|---|--|
| Size (H x W x D) in cm | 15.7 x 7.9 x 1.8 | 17.0 x 9.0 x 2.3 | 17.0 x 9.0 x 2.3 | 17.2 x 9.0 x 2.4 |
| Weight in g | 200 | 354 | 354 | 350 |
| EPOC Version | 32 | 32 | 32 | 32 |
| Screen | 480 x 160 LCD 16 grey scale (no backlight) | 640 x 240 LCD 16 grey scale | 640 x 240 TFT 16 grey scale | 640 x 240 STN 16 grey scale |
| Memory (RAM) | 8 Mb | 8 Mb | 16 Mb | 16 Mb |
| Battery Type | Rechargable Lithium-ion | 2 AA | 2 AA | 2 AA |
| Battery Lifetime ³ | 2 weeks | 4 weeks | 4 weeks | 4 weeks |
| Ports | RS232 serial port IrDA infrared port | RS232 serial port IrDA infrared port | RS232 serial port IrDA infrared port | RS232 serial port IrDA infrared port |
| Expansion Slot | - | CompactFlash type I/II card slot | CompactFlash type I/II card slot | CompactFlash type I card slot |
| Price in US\$ | 399 | 429 | 499 | 499 |
| Comments | | Additional backup battery | Additional backup battery | Additional backup battery Built-in Modem |

¹ [cmp. PSION 2000]² [cmp. Ericsson 2000]³ based on typical use profile according to the manufacturer [cmp. Ericsson 2000, PSION 2000]*Table 4.3: Specifications of sample EPOC-based PDAs*

4.1.4 Choosing the appropriate PDA

As shown above, there are many differences between the PDAs using the three operating systems. Also they focus on different types of users. Regarding to the problem analysis in Chapter 3 the following points must be covered by the device:

- The device needs a size and a weight that makes is usable at any place without problems. It must be so small and light that it fits into a shirt or suit pocket and can be used inconspicuously. Best of all would be if it fitted into a wallet, but at minimum no additional bag should be needed.
- The memory should be large enough to hold the contacts, correspondence and todo-list of a sales person.

- It should be able to dial into a corporate network when on the road. This means either a built-in modem or a link to a modem-equipped mobile phone using infrared.
- The battery should last at least for a week's extensive use.
- The user interface must be easy and quick to use. A sales person will mainly use the device for getting information about a contact, like address details or correspondence history. Also he will sometimes change existing information, e.g. the consideration factor of a contact's forecast, but he will rarely create new documents, except for short memos.

Regarding these special user profiles the Palm OS PDAs, especially the PalmVx and the TRG Pro, are the best solution. The Palm OS is much quicker to use than Windows CE, because the user interface is designed for PDAs only. Although there seems to be less memory in the Palm OS PDAs, the memory management is much more effective. As an example the application "AvantGo", an offline web service available for Windows CE and Palm OS, needs 426 kilobyte of RAM for the Palm OS and up to 869 kilobyte for the Windows CE device. The user interface of Windows CE has been adapted to the Windows standards which have been designed for the use with keyboard and mouse. Also it has lots of functions which are useless on a PDA and are made for PCs.

In bad lighting conditions a grey scale display enhanced by a backlight is more visible than a coloured one. Also a grey scale display takes less power and is lighter. This means grey scale displays should be preferred for a sales person who wants to use the device in every situation.

In general Windows CE should be preferred only for users who rely on the special multimedia features of Windows CE, like the sound recorder or movie player.

The EPOC devices are best for everybody who needs to type and edit text on the road. Because of their keyboards, the landscape format screen and the interconnectivity to the common PC word processors they are the best solutions for a mobile word processor.

4.2 PDA Software

The software needed for the integration consists of three different modules:

- The PDA front-end development software is used for developing the forms to display the information from the database and to develop the logic for processing the information.
- The PDA database is responsible for storing the information on the PDA.
- The synchronisation software is responsible for the transfer of information from the PDA to the sales information system and vice versa.

On the market are several software packages which can specifically used only for one of the three modules and some others who cover two or three modules. Also some of the software packages serve only a single PDA operating system and some can be used for more than one.

To give an overview each of the following software package descriptions are introduced by a table showing the modules and operating systems the software can be used for.

4.2.1 Native PDA Development Tools

| | Palm OS | Win CE | EPOC | Comments |
|-----------------|---------|--------|------|---|
| Synchronisation | ✓ | ✓ | ✓ | Each PDA-OS has its own software development kit, using one or more of the major programming languages, which can be used for any of the three modules. |
| PDA Database | ✓ | ✓ | ✓ | |
| PDA Front-end | ✓ | ✓ | ✓ | |

Each PDA operating system has its own software development kit (SDK). An SDK is always focused on a single major programming language and adds the PDA specific functions to the programming language. Also it contains a compiler and linker to produce an executable program for the PDA. This means that an application developed with an SDK can run as a standalone application without the need of an interpreter apart from of the ones built into the PDA operating system. SDKs can be used for developing programs which cover all three modules of the application, but can also be used for a single module interfacing with additional applications which cover the remaining ones. Using an SDK is the most flexible way to create an application, because all features of the PDA operating system can be used. On the one hand applications developed with an SDK have a high performance because no

interpreter is needed, but on the other hand the development takes longer than with a rapid-prototyping development kit.

4.2.1.1 Native Development Tools for Palm OS

Palm Inc. provides two SDKs for the Palm OS-based PDAs, one for developing applications and databases running on Palm OS, and another one for developing application-specific enhancements to the HotSync technology, called Conduits. The SDK for Palm OS, currently available at version 3.5, is made for developing applications in C or C++, for which one of the following development environments is needed:

- **Metrowerks CodeWarrior** is a commercial development tool which is preferred by Palm Inc. and is available for Windows and Mac OS
- **PRC-Tools** is a free development environment based on the Gnu Compiler Collection (GCC) and is available for Windows and Unix

For professional software development the Metrowerks CodeWarrior should be used because it is an integrated development environment which also covers aspects like version management.

For developing conduits, three conduit development kits (CDK) are provided by Palm Inc.:

- **CDK Windows C/C++ Edition** which integrates into Microsoft Visual C++ 6.0
- **CDK Windows Java Edition** which integrates into Symantec Visual Café Pro
- **CDK Mac OS Edition** which integrates into Metrowerks CodeWarrior for Mac OS

Which CDK to use depends only on the preferred platform and programming language, because they are interchangeable regarding their features.

4.2.1.2 Native Development Tools for Windows CE

Microsoft gives the developer the choice of two programming languages to develop applications for Windows CE: Visual C++ for Windows CE or Visual Basic. Both integrate into Microsoft Visual Studio which is an integrated development environment and can be used with any programming language provided by Microsoft. This means that on the one hand the developer is tied to develop all his Windows CE applications on a Windows-based PC, but on the other hand he gets the same look-and-feel no matter whether he develops in Visual Basic or Visual C++.

Also Microsoft does not differentiate between the areas of synchronisation, front-end and database. Whether to choose Visual C++ or Visual Basic depends on the kind of application to be developed. Because of their different structures Visual Basic is the better choice for developing the user interface of an application and Visual C++ is the better choice for developing core algorithms, but a mixture of both languages is also possible [cmp. Muench 2000]

4.2.1.3 Native Development Tools for EPOC

EPOC uses different SDKs for developing the synchronisation module and the application, front-end and database, running on the PDA. The synchronisation SDK provides Application Programming Interfaces (API) for the programming languages C++ and Visual Basic, whereby the developer is not tied to a development environment. For developing the applications running on the EPOC-based PDA the developer can choose between three SDKs:

- The C++ SDK integrates into Microsoft's Visual C++ development environment which has to be purchased separately
- The Java SDK can be used to create applets with any Java development environment containing a Java development kit from version 1.1.4 up to, but not including, version 2.0. These applets can run using the EPOC runtime for Java which is integrated into the EPOC operating system
- OPL is a BASIC-like rapid application development language developed only for EPOC devices. This language can be programmed directly on the device or using the OPL SDK in which case no additional compiler is required.

As with Windows CE the SDK to choose depends on the application to be developed. Using OPL is the easiest and quickest way to develop user interfaces but it does not have the performance to develop core algorithms. These should be developed with the C++ SDK and can be accessed by OPL using APIs. The benefit of developing Java applications is that they can be used on different PC operating systems without redevelopment, as long as no EPOC-specific functions have been used.

4.2.2 Rapid Application Development Tools

Rapid Application Development (RAD) tools for PDAs provide an easy user interface for developing applications running on one or more PDA operating

systems. The key fact is that they always need an additional interpreter on the PDA, because they do not produce executable programs. This also means that the performance of the developed programs is not as high as if they were natively programmed. The real benefit is that also people without much programming experience can develop applications for their PDA, which in the most cases also provide a simple synchronisation mechanism to a desktop PC.

4.2.2.1 Pumatech Satellite Forms

| | Palm OS | Win CE | EPOC | Comments |
|-----------------|---------|--------|------|---|
| Synchronisation | ✓ | | | Standard and Enterprise Edition with different synchronisation engines and licensing schemes. |
| PDA Database | ✓ | | | |
| PDA Front-end | ✓ | | | |

Pumatech Satellite Forms is a form-based RAD environment which serves Palm OS PDAs only. It provides a drag-and-drop user interface to create forms using the standard Palm OS control objects and a BASIC-like script language to add simple logic to the application. Advanced logic can be inherited using Plugins which have to be programmed in C using the Metrowerks CodeWarrior. It uses native Palm OS structures for the databases which results in high performance data access. Pumatech provides two different editions of Satellite Forms which distinguish in the synchronisation engine and the licensing scheme:

- The Standard Edition provides a synchronisation engine for a single Desktop PC only to access locally stored databases via ODBC. Applications developed using the Standard Edition can be distributed free of charge.
- The Enterprise Edition provides an enhanced synchronisation engine to access server databases, e.g. Oracle Lite, via a TCP/IP protocol and also supports the synchronisation to Lotus Domino databases using the add-on product MobileXtension. Applications developed using the Enterprise edition need runtime licenses for deployment.

4.2.2.2 GlobalWare Pylon Pro

| | Palm OS | Win CE | EPOC | Comments |
|-----------------|---------|--------|------|---|
| Synchronisation | ✓ | | | Transfers Lotus Domino databases to the PDA without programming |
| PDA Database | ✓ | | | |
| PDA Front-end | ✓ | | | |

Pylon Pro, developed by GlobalWare, is a RAD focused on Lotus Domino applications to be synchronised to Palm OS devices only. It tries to create a Palm OS application based on an existing Lotus Domino database with simulating the views and forms of that database. No programming is needed for the PDA front-end. The developer only defines which design elements of a set of databases to use on the PDA, whereby not every kind of design element, e.g. scripted buttons, can be simulated on the PDA. The result is an application on the PDA with a user interface very similar to that of the original database. Depending on the complexity of the Lotus Domino application, the PDA application has nearly the same features as the Domino application. Pylon Pro provides a conduit for the HotSync technology to synchronise the PDA database with local Lotus Domino databases. To synchronise directly with the Lotus Domino server, GlobalWare provides an additional synchronisation tool, called Pylon ServerSync.

4.2.2.3 SYWARE Visual CE

| | Palm OS | Win CE | EPOC | Comments |
|-----------------|---------|--------|------|--|
| Synchronisation | | ✓ | | The only Windows CE RAD on the market. Only simple programming logic possible without the use of native development tools. |
| PDA Database | | ✓ | | |
| PDA Front-end | | ✓ | | |

Visual CE from SYWARE, Inc. is the only pure Windows CE oriented RAD on the market at the moment. It is a development kit to create database-based applications which can synchronise to ODBC databases on desktop computers or database servers. The databases and the corresponding forms for the Windows CE device can be developed using drag-and-drop interfaces with a high level of flexibility for the form design. Also it is possible to import existing tables from an ODBC database and automatically create forms based on this table which can be used as a basis for the form development. This provides the possibility to enhance an existing application with a Windows CE front-end within hours. A programming interface has not been integrated so that it is not possible to implement programming logic except of standard functions e.g. creating reports or displaying a record with a different form. The synchronisation engine serves all ODBC compliant databases, but has no capabilities to select only a subset of an existing database table.

4.2.2.4 Feras Technologies CASL

| | Palm OS | Win CE | EPOC | Comments |
|-----------------|---------|--------|------|--|
| Synchronisation | ✓ | ✓ | | Applications developed with CASL run on Palm OS, Windows CE (BETA) and all Windows PC platforms and can synchronise with each other. |
| PDA Database | ✓ | ✓ | | |
| PDA Front-end | ✓ | ✓ | | |

CASL by Feras Technologies is a RAD providing a drag-and-drop interface for the form design and a BASIC-like script language to program the logic of the application, all based on a database engine. The difference to all other RADs on the market is that the same application can be used either for Palm OS or for Windows CE and also on any Windows 32-bit PC operating system, but the Windows CE version is still in beta testing at the moment. This means that a developer does not only create an enhancement for an existing desktop application, but he can develop his desktop application using CASL and can use this application also on his Palm OS or Windows CE device without any re-programming. Because of this architecture the provided synchronisation engine only interfaces to the proprietary database format, which is the same for all three platforms. The script language is easy to learn and provides most of the functions used for a simple application. For very specific programming issues it is possible to call external C-functions which must be separately programmed for each platform.

4.2.3 PDA Databases

| | Palm OS | Win CE | EPOC | Comments |
|-----------------|---------|--------|------|---|
| Synchronisation | ✓ | ✓ | ✓ | Slimmed down versions of relational database management systems with engines for all PDA platforms. |
| PDA Database | ✓ | ✓ | ✓ | |
| PDA Front-end | | | | |

The backend database can either be developed using a native development environment as described in Chapter 4.2.1, or an additional PDA database engine can be used. The PDA databases on the market are slimmed-down versions of relational database management systems (RDBMS), developed by the market leaders in this area. For use on PDAs these RDBMS are reduced to the core engine which only provides APIs for running SQL queries. Compared to the versions running on Windows-based PCs or database servers they need only a very small amount of memory.

Both major RDBMS manufacturers, Oracle and IBM, provide such PDA databases. They are called Oracle8i Lite and IBM DB2 Everywhere and both support all three PDA operating systems. Either database can be accessed through an API using the C or C++ based native development environment of each PDA operating system.

4.2.4 PDA Synchronisation Engines

Instead of developing a synchronisation tool for each PDA application a synchronisation engine can be used. The synchronisation engine consists of two parts: a synchronisation server running on a network server and the client running on the PDA.

These engines serve most of the PDA operating systems and usually provide additional features for security and management of the mobile devices.

4.2.4.1 Oracle iConnect

| | Palm OS | Win CE | EPOC | Comments |
|-----------------|---------|--------|------|---|
| Synchronisation | ✓ | ✓ | ✓ | Supports Oracle and Oracle Lite databases only. |
| PDA Database | | | | |
| PDA Front-end | | | | |

Oracle iConnect is a synchronisation engine focused on the Oracle Lite database. This is the only kind of PDA database which can be handled by iConnect but the three major PDA operating systems can be used. On the server side iConnect synchronises with any Oracle database of version 7 and above. Which subsets of the server database to synchronise can be defined using SQL statements. Apart from the synchronisation iConnect can deploy applications and files to the PDAs. For connecting the PDA to the server a TCP/IP connection is needed, no matter whether LANs, WANs or wireless networks are used.

4.2.4.2 IBM Mobile-Connect and Extended Systems XTNDConnect

| | Palm OS | Win CE | EPOC | Comments |
|-----------------|---------|--------|------|--|
| Synchronisation | ✓ | ✓ | ✓ | Supports ODBC databases, Microsoft Exchange and Lotus Domino |
| PDA Database | | | | |
| PDA Front-end | | | | |

Extended Systems XTNDConnect and IBM Mobile-Connect are both based on the same technology which was developed by Extended Systems and licensed by IBM,

but XTNDConnect exists as version 2.4 and Mobile-Connect as version 2.2. The main differences between the two versions are that MobileConnect is not available for EPOC PDAs and that XTNDConnect encrypts the data transmitted from server to PDA which means a higher level of security. Both products can be used for database synchronisations and also for administrating PDAs, e.g. regular device backup or software installations. Synchronisations can be set up for any kind of databases on the PDA using different plugins. The plugins shipped as standard can handle any native database format and DB2 Everywhere databases on any PDA and additionally Satellite Forms databases on the Palm OS devices. Additional plugins can be developed in C++ if needed. On the server side the synchronisation engine can handle any ODBC database, Microsoft Exchange and Lotus Domino databases. Aside from database synchronisation the synchronisation engine can also replace the engines shipped with the PDAs to synchronise emails or calendars with the above groupware servers. The PDAs can connect to the synchronisation server in different ways as shown in Figure 4.2 whereby the only prerequisite is an existing TCP/IP-connection to the server.

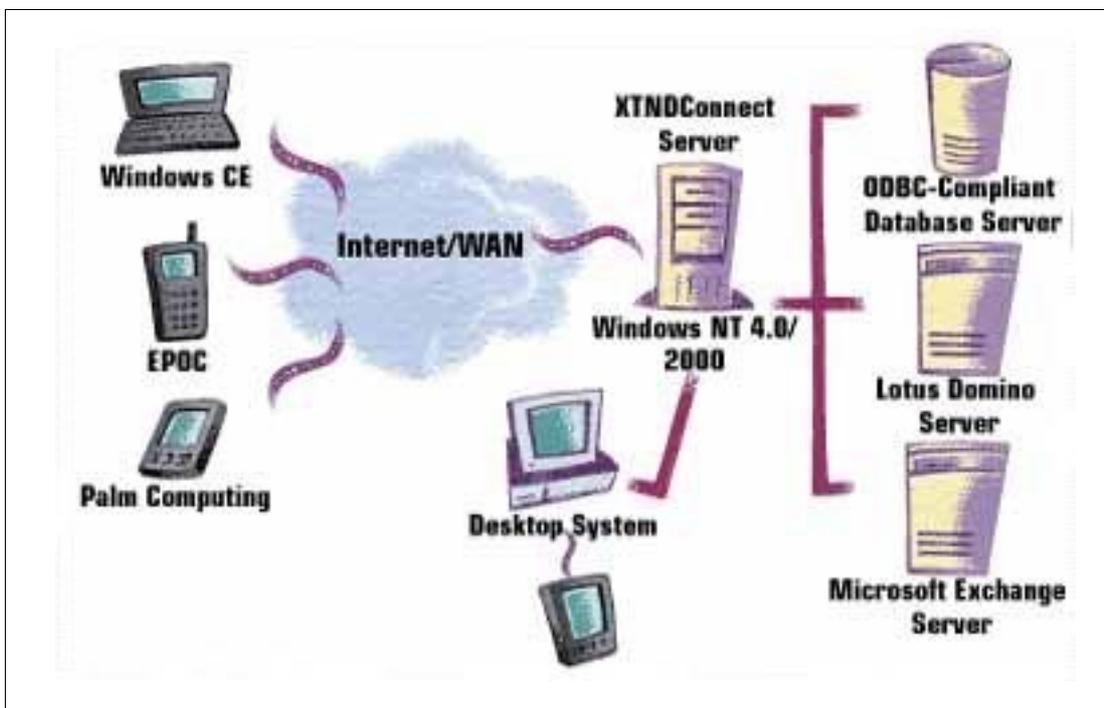


Figure 4.2: XTNDConnect Architecture [ExtendSys 1999a]

4.2.5 Architecture decision for the PDA Application

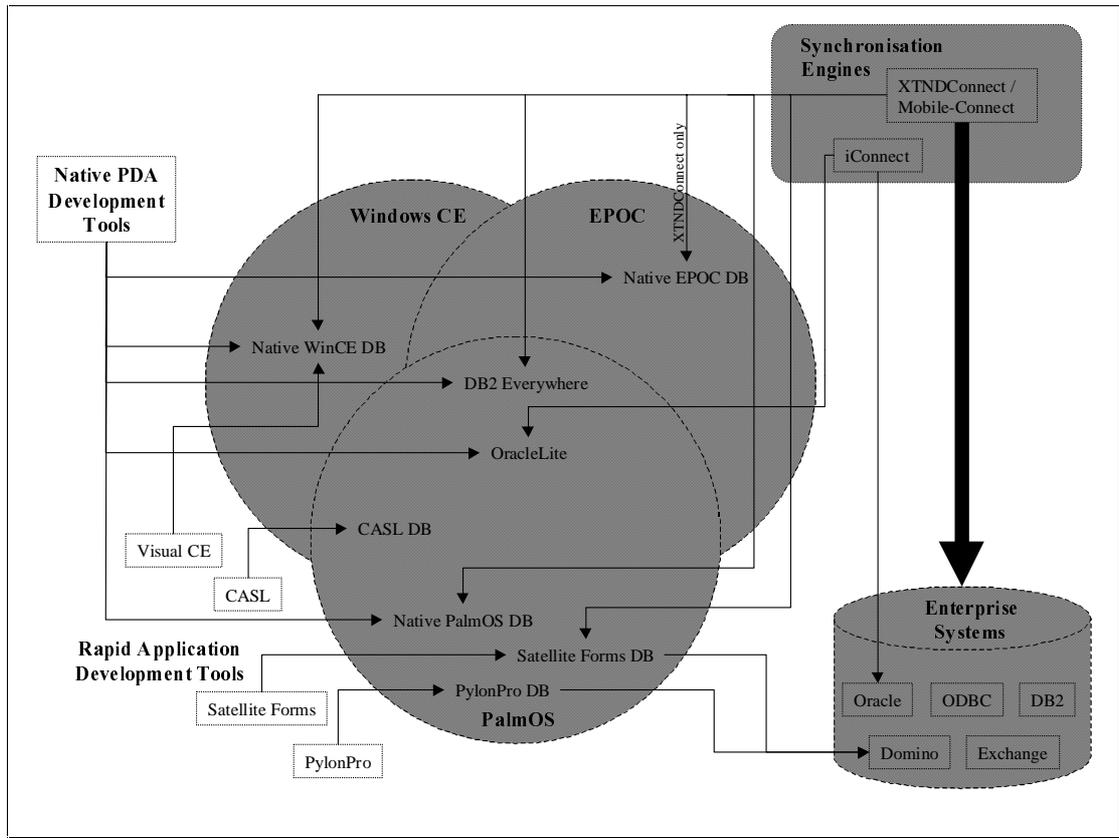


Figure 4.3: Interfaces of PDA Application Modules

When building an architecture the main focus should reside on how to synchronise the information between the PDAs and the Enterprise systems. Except for a system which only needs to support a small number of users all using the same PDA and the same applications, a synchronisation engine is indispensable. The engine should be the centre of attention for the architecture around which all PDA applications are built. At the moment XTNDConnect is the best choice regarding security features, flexibility in PDA and Enterprise databases and administrative issues for the PDAs. Whether to choose a platform-independent database depends on the performance needed and the variety of PDAs used. In most cases a natively developed database is more performant than the platform-independent one but has to be developed for each single operating system.

Whether to use a native development environment or a RAD tool depends on the features and performance needed. Simple applications can be developed with a RAD whereby the performance will be not as good as with a native developed application and some features of the PDA will not be available.

Also it must be considered how the three modules can interface with each other.

5 Concepts for the Integration

This chapter shows the concepts which have been used to implement the integration of PDAs into a process-driven Enterprise sales system. The main issues when using PDAs are on the one hand how to handle the dataload regarding the reduced amount of memory of the mobile devices, and on the other hand how to handle the workload focussing on the capacity of a PDA processor. Also there are different considerations when implementing a user interface for a PDA than for a PC.

5.1 Handling the Dataload

Handling the dataload on the PDA is the one of the aspects when integrating PDAs into Enterprise systems, because the storage capacity of a PDA is extremely small compared to the storage capacity of a database server. This means that considerations must be given to reduce the amount of data stored on the PDA.

5.1.1 Three-step Data Volume Reduction

This chapter describes a way to reduce the amount of data stored on the PDA. It divides the reduction into three steps which are carried out during the process of synchronisation. A database can be regarded as a three dimensional object. The first dimension defines the number of records of that database. The second dimension is the number of fields of each record and the third dimension is the amount of data which is stored in each field of each record. The Three-step Data Volume Reduction tries to reduce the amount of data stored on the PDA by reducing all three dimensions of such a database cube.

In the first step the axis counting the number of records is considered. This is also the step which provides the highest level of reduction. One possibility could be to mark the records on the server manually and synchronise them. On the one hand this is the most precise way of synchronising the correct records because there is no record on the PDA afterwards which is not needed. On the other hand the manual marking is very time-consuming and there is a definite likelihood of missing a record. Another problem of this method is that relationships between records are not considered. For example having the correspondence history of a contact without the contact itself could force errors when displaying the information on the PDA. Therefore a different method should be used. This method is not more precise than manual marking but it

is much more error-proof. Based on a specific formula the records which should be synchronised are selected from the server database automatically. How this formula is put together and on which rules it is based is described in Chapters 5.1.2 and 5.1.3. The second step of data reduction regards the number of fields of the records. Process-oriented systems usually need lots of fields which contain process-related information, e.g. process status, task status, task member lists etc. Also there are fields which are not used by the process engine but just contain calculated values which are stored so that they do not need to be calculated every time the record is used. Because a server database does not have any capacity limits or only virtual limits which are not reached by the system this can be useful for enhancing the performance when accessing the data. When using a PDA database which is very limited in its capacity such fields should not be stored in the database but only be calculated on demand. Also any process-related field which is not needed on the PDA should not be transferred to the PDA database for saving space. This kind of reduction is done when mapping the fields of the server database to the fields of the PDA database. So the fields are not selected during the synchronisation process at the time of each synchronisation, but generally for all future synchronisations when configuring the synchronisation engine.

Step 3 is to reduce the amount of data in each field. This kind of data volume reduction must be carried out very cautiously because in most of the cases reducing the amount of data in a field also means losing information. Or even worse it can corrupt the information stored in that field. But there are some cases in which this method can be used without losing or corrupting information:

- Date fields containing dates which are known by the user to be in the near future or past can be reduced by removing the year.
- Large binary objects, e.g. files, which cannot be used on the PDA can be removed from the PDA database but notification about this truncation should be given to the user.
- Fields containing keywords can be shortened by using abbreviations for these keywords. In any case these abbreviations must be well known to any user.

Depending on the type of application there are more additional possible truncations to be thought of, e.g. cutting phrases. The only fields which never should be truncated are fields containing numerical values because this would always corrupt the value. How this step of data reduction is carried out should be defined by the

administrators and developers of the system when the system is set up. It is possible either to prepare the information stored in the server database so that it remains there in the reduced format or to carry out the reduction during the synchronisation process. Which method to use mainly relies on the capability of the synchronisation engine to modify data during synchronisation.

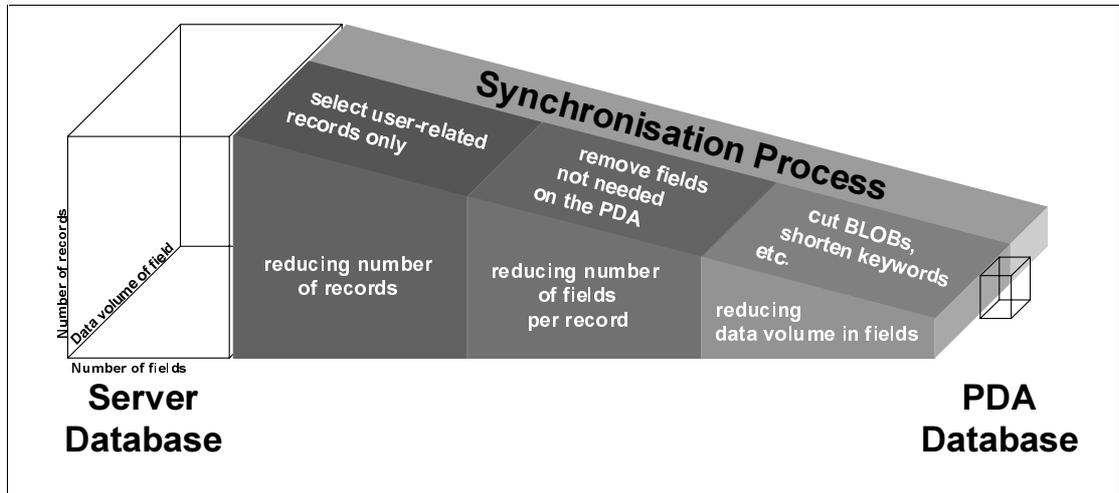


Figure 5.1: Three Step Data Volume Reduction

5.1.2 Predefined and User-defined Data Selection

As mentioned in Chapter 5.1.1 the most effective method when reducing the data volume on the PDA is to reduce the number of records by using a formula for selecting specific records. This formula can either be predefined by the synchronisation engine or defined by the user and used by the synchronisation engine.

Predefined formulas are designed once and used for every user of the system. It can be described as a push-technology because the user has no influence which information he synchronises to his PDA. The real benefit here is that an administrator can make sure that each user gets the information he should get. This kind of technology should always be used when transferring information to the PDA which is needed by the PDA-system to run properly, e.g. setup-information.

User-defined formulas are defined by each user according to their own desires. This kind of pull-technology transfers control regarding the information to be synchronised completely to the user. It should be used any time where it is not possible to define a selection which suits the needs of all users. When using the user-defined selection the following aspects have to be considered:

- Complexity of the formula

Formulas which are used for selecting records from databases mainly consist of concatenated equations in which single fields are compared to a given value or value-range. This kind of formula language is too complex for most of the users. Also a user should not need to know the definition of the fields of the database. This means that the user should not be forced to learn a complex formula language for selecting his information. The easiest way for the user is to provide him with a profile containing different business rules as described in Chapter 5.1.3. In the background an algorithm generates the real formula based on this profile.

- Completeness of the formula

A benefit of using a formula for data selection instead of using manual data selection is that it can be much more error proof. This means that the algorithm which generates the real formula from the user profile should be that intelligent that it recognises relations between different records and adapts the formula to them automatically, e.g. not transferring the correspondence history of a contact without transferring the contact itself.

- Entry-point for the user

The user should not be forced to use a system for entering the profile different from the one he already works with. This means that the entry-point for the data selection profile should either be part of the front-end of the server-based system or part of the PDA front-end. A combination of both, setting up the profile on the server-based system and afterwards being able to change the profile on the PDA, is also suitable, but it has to be regarded that if the business rules of the profile change they have to be redesigned for both platforms.

5.1.3 Business Rules for the Data Selection

As described in Chapter 5.1.2, a profile for data selection should be built from several business rules. A business rule describes a specific set of data in natural language which can be transferred into a data selection formula. The use of natural language makes this rule easy to understand for the user but it could be that the affiliated formula is very complex and sometimes also has to adapt to specific parameters. An example for a process-driven system would be the business rule “My Tasks for today and the next week”. Every user understands what this expression

means, but the formula in the background is the very complex because it has to look for several fields containing usernames and date-ranges and it must be adapted to the current date every time it is used.

The best way to find out such business rules is to interview the users. Asking them how they categorise the information they would like to synchronise will provide a rough set of natural language business rules. These must be analysed to remove business rules which would direct to the same set of data and to complement them with additional rules to make sure that every set of data can be reached by the rules. Afterwards an algorithm is needed which transfers each of these rules into parts of a data selection formula and concatenates all parts to get the whole formula.

5.2 Handling the Workload

Concepts for workload management in the context of PDA integration do not cover the aspects which are known as workload management of Enterprise systems like job scheduling and resource management between different servers. Workload management in this area means how to transfer as much processing time from the PDA to the server to increase the performance of the PDA application. Additionally decreasing the workload on the PDA means to increase the PDA's battery lifetime because the power management of a PDA turns off the processor of the PDA as soon as possible no matter whether the display is turned on and the PDA is in use. The first part of this chapter focuses on the workload caused by a process-driven system and how this special kind of workload can be reduced on the PDA. The second part focuses on workload issues which do not occur in process-driven systems only but in any form-based system.

5.2.1 Server-based Routing

Task routing in process-driven applications consists of several steps but the two main steps are determining the next task or tasks and determining the editor or editors of that task. Several additional actions can be done during the routing of a task like computing specific data or triggering actions in external systems but these are optional and not always needed. Regarding a predefined process the next tasks in a process can be determined from the process definition stored in a repository. The editors of the next tasks are also defined in the process definition but they are not always certain at the time when the process is designed. This means that groups or

roles can be used as editors and the definite editors are determined from a repository containing the organisational structure when routing the task.

For ad-hoc processes the user enters the tasks during the runtime of the process at the same time as he defines the editor of the next task during the runtime. For defining the editor the use of groups or roles is also possible and the process engine will determine the real editors when initialising the ad-hoc task.

For task routing two different concepts are possible:

- Client-based Routing

When using client-based routing the process engine tries to determine the next tasks and editors directly after the user tells the process engine to start the routing. The routing operations itself are carried out by the client application and errors during the routing process are reported immediately to the user.

- Server-based Routing

Server-based Routing means that the user sends a request to the process engine to start the routing of a task but the routing process is not always started immediately. The process engine runs in the background of the application on a specific schedule and routes all tasks which have been requested to be routed. Errors occurring during routing are not displayed to the user immediately.

There exist several reasons why server-based routing is the only choice when using PDAs:

- The processing performance of a PDA is much lower than the performance of a PC. This means that such complex operations like task routing degrade the performance of the PDA application too much
- When using server-based routing the repositories containing the process definition and the organisation structure are not needed on the PDA because only the process engine running on the server accesses them.
- The complex algorithms of the process engine need not to be implemented on the PDA, because the existing server engine is used. This both reduces development time for the PDA application and also ensures that no memory is wasted for the program-code on the PDA.
- Regarding the fact that in a process two successive tasks are usually worked on by different editors there is no need to determine the editor of the next task before it is synchronised to the server. The PDA is used by a single person only and

there is no benefit for that person to have a task on the PDA he does not have to work on.

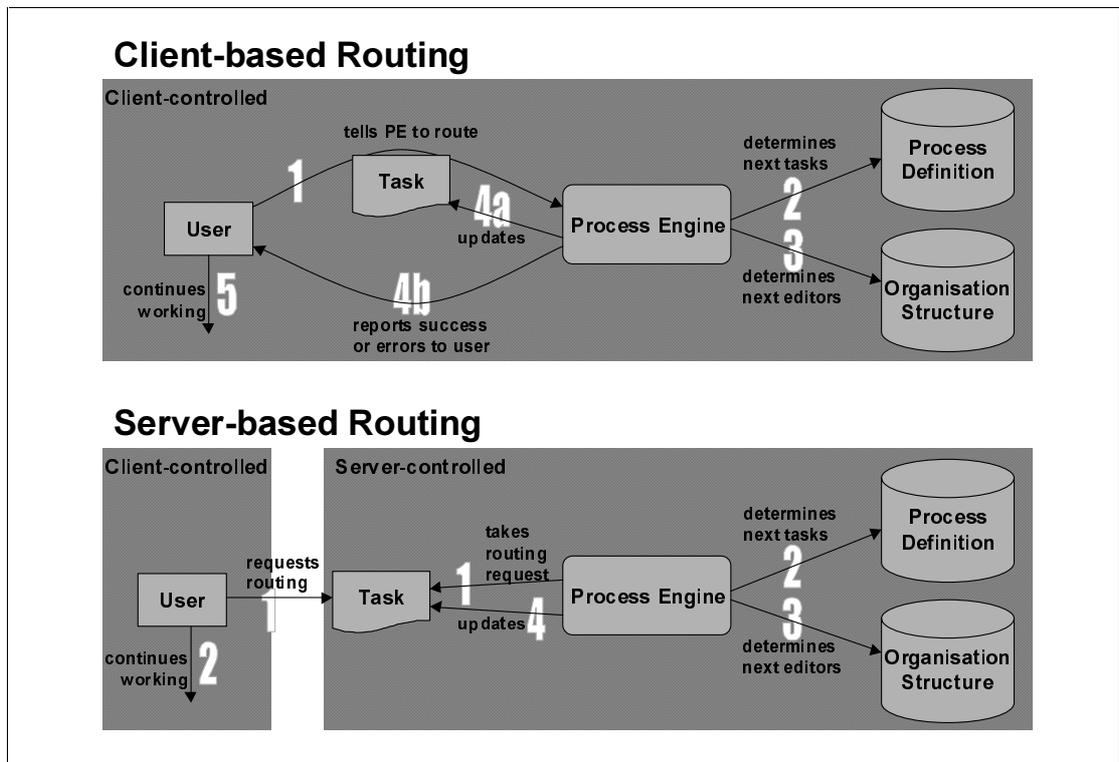


Figure 5.2: Client-based Routing vs Server-based Routing

When implementing server-based routing in the PDA environment, the PDA acts as a workflow trigger. This means that if a user wants to route a task to the next editors he only marks that task as to-be-routed. This mark is transferred to the server during synchronisation and recognised by the process engine which initialises the routing. Two aspects have to be considered when implementing this concept.

- At some stages in a process it is necessary that the user decides the routing direction of a task himself. These routing options must be available on the PDA at the point where he marks the task for routing.
- If an error occurs during the routing of a task it is necessary that the process engine notifies the user in a way that he can resolve the error and re-routes the task on the PDA.

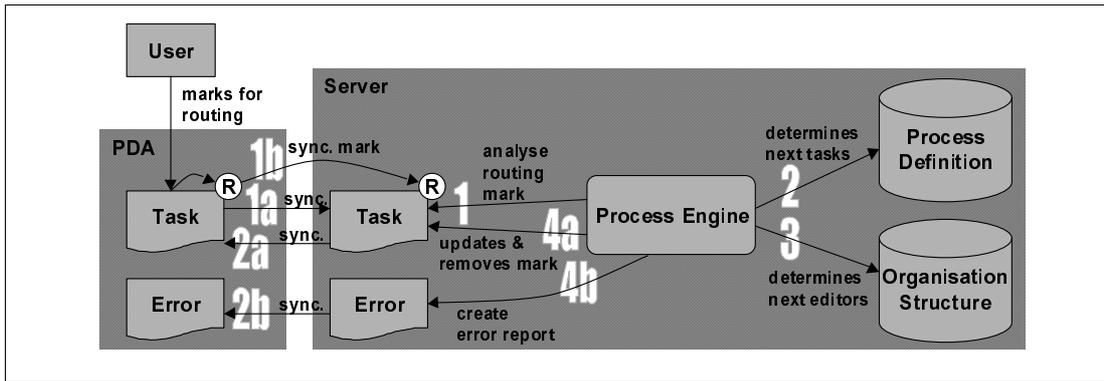


Figure 5.3: Server-based Routing for PDAs

5.2.2 Server-based Validation

Validation is used in form-based applications to make sure that a form has been filled out completely and correctly. Also validation can be enhanced in a way that based on the values entered in the form an action is triggered, e.g. computing an additional value and storing it. Validation usually takes place automatically when the information entered in a form is saved but also can be triggered on user request to check whether a form has been filled out properly. Additionally the validation logic can differ based on the state of the information displayed with the form. It is possible that in a process-driven environment the validation is different for each task of a process also if the same form is used to display the information. Because validation often accesses several values to generate a new one, problems can occur when using it on a PDA. The concept for reducing the dataload described in Chapter 5.1.1 has the effect that the number of fields on the PDA are different than in the server database and also the values in the field can be truncated or reduced to a space saving format. This means that if the validation logic of the server is used on the PDA the results can be wrong because of missing values. An additional problem causes again the performance of the PDA processor. Complex validations will slow down the PDA application and make it unusable. To remove this kind of workload from the PDA and to avoid the problems mentioned above a concept called server-based validation should be used.

Server-based validation means that the validation of a form is not carried out on the PDA but a background process on the server handles the validation directly after the synchronisation between PDA and server. The benefits for using this concept with PDAs are obvious:

- It reduces the workload on the PDA and transfers it to the server. Because of the higher and easy extendable performance of the server it has no negative effects.
- The validation need not to be implemented for the PDA which reduces the development effort for the PDA application. The PDA application need not to be updated every time the validation logic changes.
- No process-specific information must be transferred to the PDA to carry out the validation.

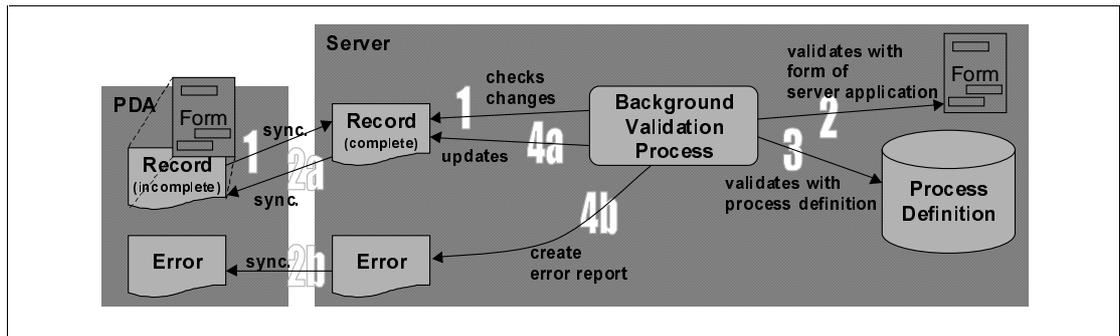


Figure 5.4: Server-based Validation on PDAs

5.3 PDA User Interfaces

The small size of the screen of a PDA combined with a pen as an input-device brings up some new aspects when designing a user interface for a PDA. Additionally the user requirements to the PDA user interface are different than to a PC system. There are two major issues to be regarded when designing the user interface for the PDA

- Reading vs. editing information

PDA devices are mainly designed for viewing information than for creating and editing them. This means that either each form of the user interface but also the flow of control between the forms should be optimised for reading information. When there is the need to create or edit information the user interface must support the user with methods to speed up data-entry.

- Information access speed

A major advantage of the PDA against a notebook is the reduced start-up time. This advantage should be continued by the user interface in such a way that data is reached in the shortest possible time. The actual elapsed time has to be considered, but more important the number of taps to get to the needed information should be focussed on when designing the user interface.

The following methods for user interface design are focussed on the above aspects:

- Use of modes

When designing a user interface for a PC-based system the use of modes should be avoided because it forces the user to recurring sequential actions. [cmp. Keil-Slawik 1999]. Because the focus for a PDA user interface is more to read information than to edit it, combined with a very small screen, the use of different modes for editing and reading data is reasonable if these modes are switchable with a single tap. A form designed for reading only does not need any controls to perform actions, except for switching to the edit-mode form and also field labels needed in an edit-mode form can be mainly removed. Both saves space on the display which can be used for displaying the information instead.

- Checking the frequency of functions

The frequency of use of a single function on the PDA differs, e.g. checking a schedule is done several times a day, but entering a new appointment to the schedule is done less frequently. Regarding these different frequencies the user interface has to be optimised in a way that the most frequent functions can be reached with the fewest taps possible, e.g. checking today's schedule is done with a single tap on a Palm OS device [cmp. Palm 1999, p. 34-35].

- Use of dialogs

The use of pop-up dialogs to enter information should be minimised because they increase the number of taps. They should only be used if they can speed up entering the information, e.g. picking a date from a calendar view dialog instead of typing it in [cmp. Palm 1999, p. 36-37].

- Overview forms vs. detailed view forms

Except of dividing forms into edit-mode and read-mode forms they can also be categorised into overview and detailed view forms. An overview form shows a list of all or a subset of the records of the database with displaying the most important information of each record. Detailed view forms display all information of a single record. When designing overview forms the main focus should lie on which information of a record to display. The main goal when designing such a form should be to display the most frequently needed information of a record so that the user is not forced to dig into the detailed view form.

6 Integration Prototype

A prototype has been developed as part of a project within the telesales department of SYMANTEC (UK) Ltd., to develop a practical solution for integrating PDAs into a process-driven Enterprise sales system. SYMANTEC is an international software development company with a product range including security software and PC support software.

The overall aim of the project was the development of a sales information system based on the PAVONE Espresso technology and the integration into the current IT environment of SYMANTEC. A prototype PDA integration is an additional component of the project.

Because the majority of the sales people at the SYMANTEC sales department use Palm OS-based PDAs for their personal information management, the prototype has been developed exclusively for this kind of PDA. However, the expansion to other kinds of PDAs has been considered as a deciding factor when choosing the components for the prototype.

This chapter will describe the features and the technical development of the prototype.

6.1 The PAVONE Sales System

PAVONE Sales is a process-driven sales information system based on the PAVONE Espresso technology. It has been developed to fulfil the needs of three different user groups.

- **Sales people** use it for customer relationship management. They store lead information, correspondence, quotes, forecasts and orders in a shared Lotus Domino database. Using the Espresso workflow engine they use structured processes for qualifying leads and ad-hoc processes for assigning tasks based on these leads.
- **Sales managers** use it as a tool to manage their sales teams. An interface to ShowBusiness Cuber & KAS has been developed, to produce reports and charts for monitoring the sales force effectiveness.
- The **marketing department** uses it as a data pool for targetting the right addressees for their marketing campaigns and as a management tool to measure the efficiency of their marketing campaigns.

Contact information and order information can be interchanged with financial systems' relational databases.

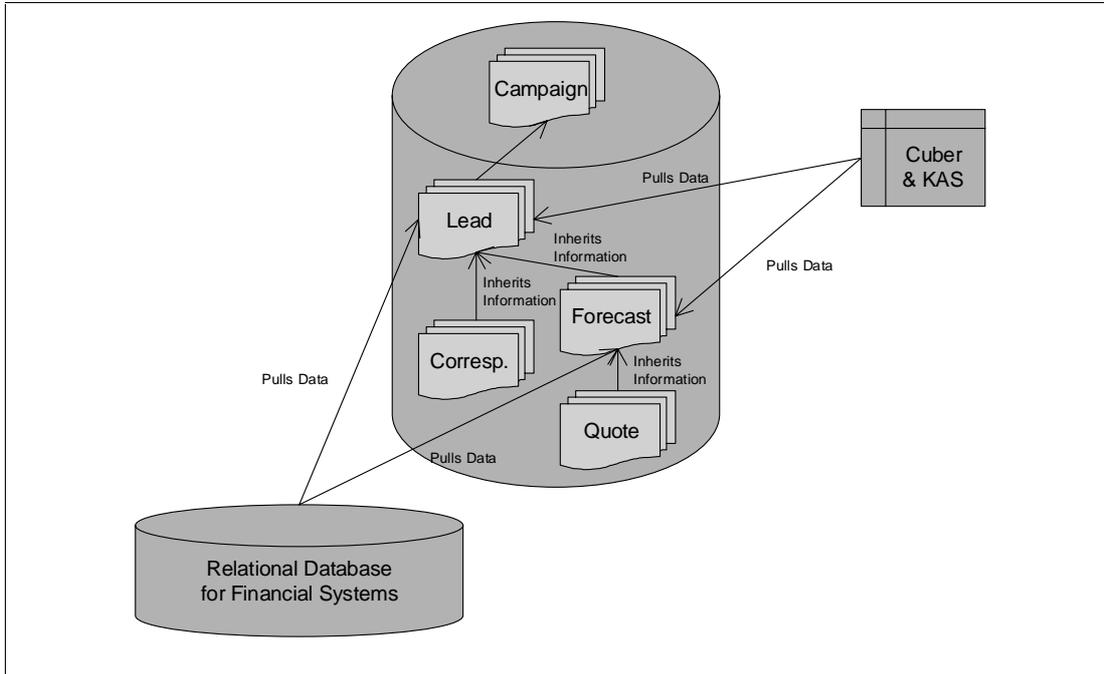


Figure 6.1: Architecture of the Sales Application Database

PAVONE Sales consists of four databases and two external graphical tools:

- The **Sales Application Database** is the main workplace for all users. It contains all sales-related documents: leads, forecasts, memos, letters and campaigns. These documents can be routed through predefined processes or be used as base information for assigning ad-hoc tasks to users.
- The **Organisation Database** contains the organisational structure, with details like persons, departments, groups and roles. The process engine needs this information for assigning tasks to the right person at the right time.
- The **Settings Database** contains keyword lists, letterheads, document templates and pricelists. It serves as a repository for the Application Database which inherits the components during the runtime.
- The **Process Definition Database** is also a repository. This repository contains the routing structure for all predefined structured processes to be used in the Application Database, for example the lead qualification process.
- The **OrganizationModeler** is an external graphical tool for displaying and modelling the organisational structure in an intuitive and user-friendly way. It uses the Organization Database as the data storage.

- The **ProcessModeler** is an external graphical tool for displaying and modelling the routing structure of predefined structured processes which are stored in the Process Definition Database. It can be also used for simulating a newly defined process to identify bottlenecks or deadlocks in that process.

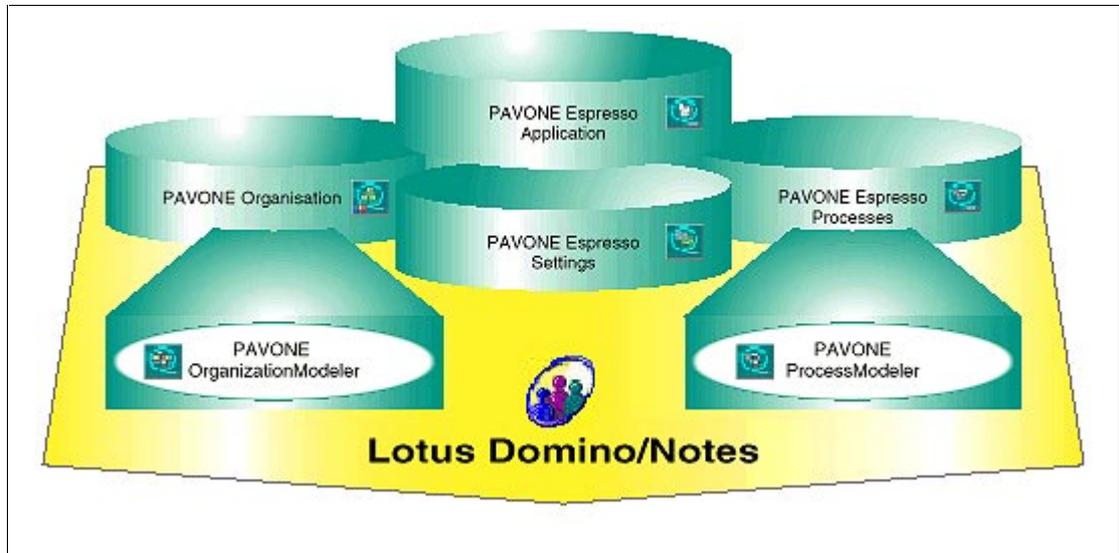


Figure 6.2: The PAVONE Sales Architecture [PAVONE 1999, p. 3]

6.2 Tools and Architecture

The following tools have been chosen for implementing the prototype:

- PDA Front-end Development:
Metrowerks Codewarrior for Palm OS, Release 6
- PDA Backend Database:
IBM DB2 Everywhere, Version 1.2
- PDA to Lotus Domino Synchronisation:
Extended Systems XTNDConnect Server, Release 2.4

There are three main reasons why this combination has been chosen:

- The first reason is the ability to expand the system to any kind of PDA with as little effort as possible. XTNDConnect 2.4 as synchronisation server can be used with Palm OS, Windows CE and EPOC based PDAs. Also DB2 Everywhere is available for Palm OS and Windows CE whereby EPOC is supported in the current beta version. This means that only the PDA front-end has to be redeveloped for each kind of PDA, but as mentioned in Chapter 4, there is no

development platform available yet which can be used for all three kinds of PDAs.

- The second reason the possibility to implement a user-defined data selection as described in Chapter 5.1.2, with as little administrative effort as possible. XTNDConnect 2.4 was the only tool to provide this flexibility. As an additional benefit it is possible to use the Personal Settings document of the Application Database as a data selection profile, so that no additional user training is needed for the use of this profile.
- The third main reason concerned security and administrative issues. XTNDConnect is the only tool which provides an end-to-end security no matter whether the synchronisation is done using a LAN, a modem connection or the Internet. It also provides scheduled data backups and an easy way of updating software components on the PDAs.

The combination of the chosen tools forms a flexible and expandable solution with the ability to implement all concepts described in Chapter 5. The result is an architecture as shown in Figure 6.3.

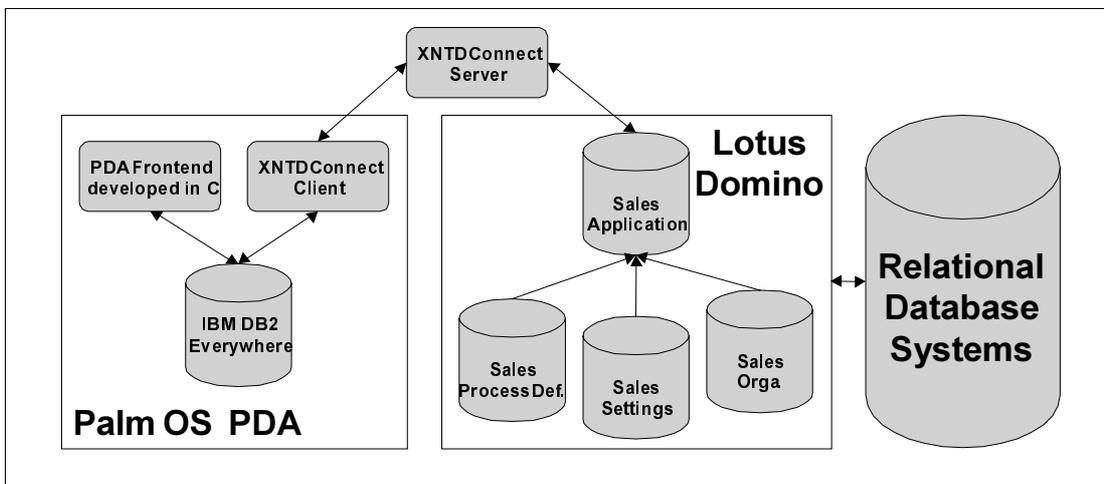


Figure 6.3: Architecture of the Prototype

6.3 The PDA Backend

The backend databases on the PDA have been implemented using IBM DB2 Everywhere and consist of six different tables:

- **ES_Address** contains the contact records.

- **ES_Corresp** contains all kinds of correspondence information, like letters, memos and received documents. Each record is related to a specific record of ES_Address.
- **ES_Forecast** contains the forecast information and the records are also related to a record of ES_Address.
- **ES_Orga** contains organisational information needed for assigning ad-hoc tasks.
- **ES_Process** contains information which is needed to initialise a new process.
- **ES_Setting** contains information about keyword lists used by the front-end.

The first three tables contain the information the user works with. He can read, edit and also add records to these tables. Additionally each record stored in one of the first three tables contains information about the current workflow status of the related document in Lotus Notes which is needed to implement the workflow features of the PDA front-end. The latter three tables are read by the system only and cannot be changed on the PDA by the user.

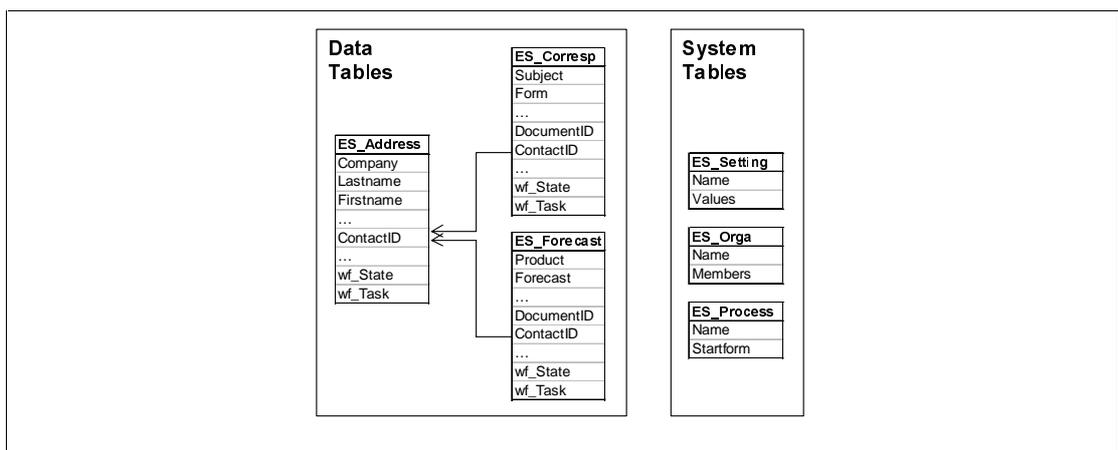


Figure 6.4: DB2 Everywhere Data Structure

6.4 The PDA Front-end

The PDA front-end has been developed in native C using Metrowerks Codewarrior 6. Using a native PDA development tool was the only possibility to implement the features described below with having enough flexibility to enhance the system by additional features.

- Contact and correspondence management

The user can read, edit and create new contact records on his PDA. He can view the correspondence history of his contacts, whereby the following document

types are supported: Memo, Letter, Received Document. Also he can edit each of these documents and create new ones.

- Forecast management

The user can read, edit and create forecast records which are either displayed in the correspondence history of a contact or additionally as a separate personal forecast overview.

- Workflow management

An overview of the current tasks and the documents to be reviewed is provided. These tasks can either be completed or a new review date can be assigned. Based on any kind of document new ad-hoc tasks can be initialised by the user as well as predefined processes can be started.

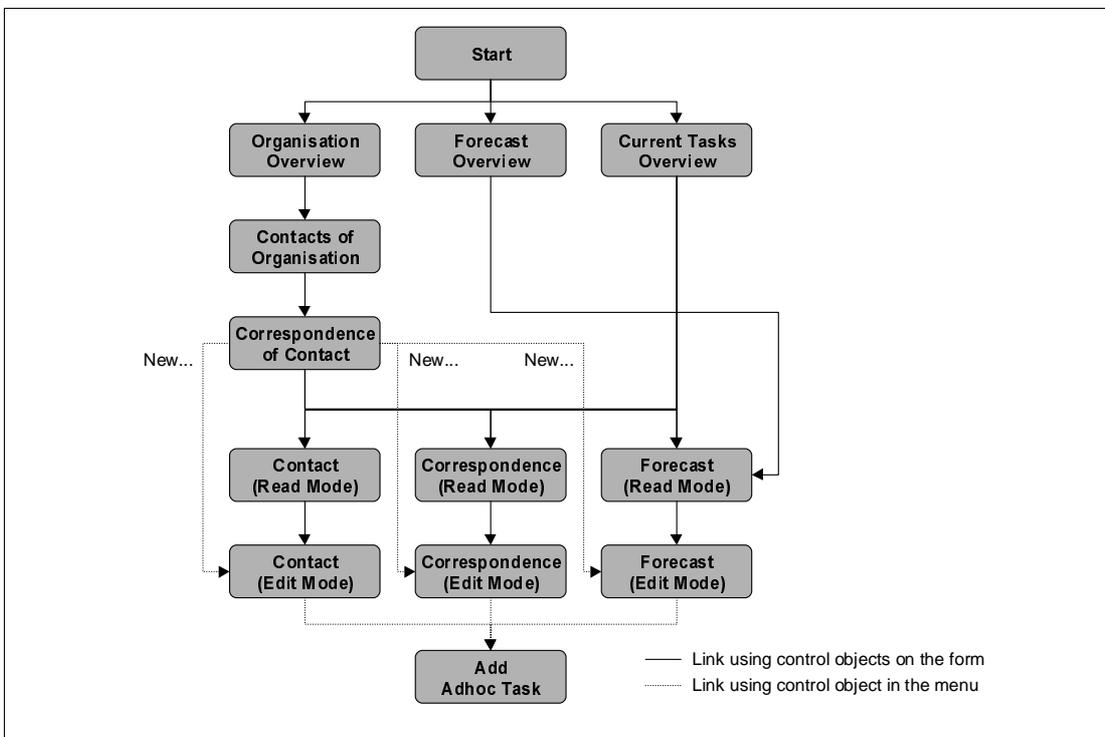


Figure 6.5: PDA Front-end – Form Structure

The Palm OS uses forms to display information and control objects. Control objects can be placed directly into the form or in the Palm OS menu. Placing control objects into the form has the benefit of decreasing the time and the number of taps to reach a specific function or piece of information. Placing them in the menu means that more space on the screen can be used for displaying information. As a compromise, between quick data access and displaying as much information as possible on one screen, all control objects needed for navigation and editing have been placed on the forms whereas controls which provide additional features have been placed in the

Palm OS menu. Figure 6.5 gives an overview about the different forms and how they are linked together.

To realise the user interface concepts described in Chapter 5.3 three main types of forms have been used:

- **Overview forms** contain a list showing information of a subset of the records from one or more tables.
- **Read-Mode forms** contain all information of a record as structured text without separating fields and labels. A scrollbar has been used to display all the information on the small screen.
- **Edit-Mode forms** contain all information of a record and give the user the ability to edit the content. The form is separated into fields and labels and is divided into separate pages.

Only two forms do not fit into this scheme: The **Start form** does not display any record information. It is used as a start page to navigate to different areas of the software. The **About form** displays the version information of this software.

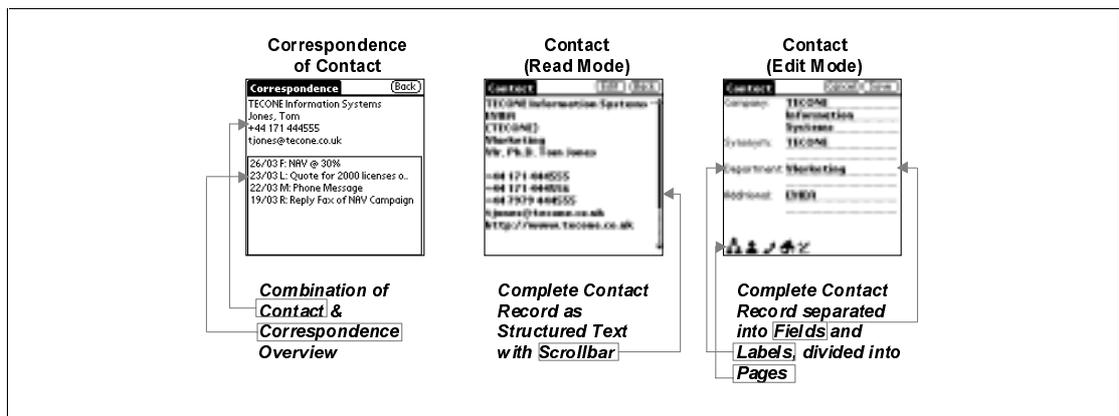


Figure 6.6: PDA Front-end – different Form Types

The main focus when developing the forms was to design a fast and intuitive interface where no user training is needed. The user needs only three taps to view the telephone number, email address and the correspondence history of a contact, another tap to read all contact information or one of his correspondence documents, and two further taps to assign a new ad-hoc task based on the contact or correspondence document. Regarding the user interface concepts described in Chapter 5.3, the front-end has been optimised for viewing information, not for editing. However, when editing a record the user is provided with popup-lists for quick and easy data entry.

6.5 Enhancements of PAVONE Sales

After a description of the additional and modified elements of PAVONE Sales it will be shown how the concepts of server-based routing and server-based validation have been implemented.

6.5.1 Enhancements of existing Forms

Each form in the Application Database which is used for creating documents to be synchronised with PDAs needs an additional subform. This subform contains several fields which are needed for the synchronisation in two ways.

First there are fields which contain specific parameters of the Notes document which are usually not stored explicitly in a field. Because the synchronisation tool can only access fields, these parameters have to be extracted from the document and have to be stored in a field.

One example is the field "DocID". It contains the document unique ID of the document which can be extracted from the document using the Notes function "@DocumentUniqueID". This ID is used either as a unique key in the DB2 Everywhere tables or for cross-referencing between contact records and correspondence or forecast records.

The second kind of fields computes values from existing fields to reduce the amount of data stored on the PDA (cmp. Chapter 5.1.1, step 3 of the data volume reduction model). In most cases this is a compromise between gaining memory capacity on the PDA and losing information whereby the main focus should be on losing as less information as possible.

One example is the field "PDA_Subject" which tries to shorten the existing subject of the document. The subject of a forecast document is usually computed as "Tracking: <ProductName> at <ConsiderationFactor>%" as long as the consideration factor is below 100%. This means that the subject is usually too long to be displayed in one row of a list on a PDA form. To reduce the amount of data on the PDA a shortened subject, like "<ShortProductName> @ <ConsiderationFactor>%" is computed for this field and sent to the PDA.

6.5.2 User-defined Data Selection

The Personal Settings form of the Application database has been enhanced with an additional subform. This subform holds information about which information the user wants to synchronise with the PDA. Via a user-friendly interface the user selects which contacts to synchronise by selecting the companies manually or using a formula. Correspondence and forecast documents can be selected by date range whereby forecast documents additionally can be selected by their consideration factors. In addition, the user can select how to handle the tasklist and whether he wants to be able to initialise processes on his PDA. The user interface of this subform is shown in Figure 6.7.

The screenshot shows the 'PDA Settings' subform with the following options:

- Contacts:**
 - Synchronise all companies containing:
 - Like: IBM, Symantec, ...
 - as Company Name
 - as Company Synonym
 - as Company Name or Synonym
 - Synchronise all companies which fulfil the following selection:
 - @Contains(ProductIDopportunity,"Morton Ankihus")
- Correspondence:**
 - Do not synchronise my correspondence
 - Synchronise the correspondence of the contacts defined above
 - Synchronise the correspondence of the following companies:
 - Document Types:**
 - Mails
 - Letters
 - Received Documents
 - Synchronise All
 - Synchronise by date created
 - The last three months ...
- Forecast:**
 - Do not synchronise my forecast
 - Synchronise all my forecast
 - Synchronise my forecast for the contacts above
 - Synchronise all forecast of the contacts above
 - Synchronise my forecast of the following contacts:
 - Synchronise all forecast of the following contacts:
 - Forecast date between Three months in the past ... and Three months in the future ...
 - Consideration Factor between 20% ... and 30% ...
- Workflow:**
 - No workflow specific synchronisation
 - Synchronise my current tasks
 - Synchronise my current tasks and tasks with a review date until
 - The next visit ...
 - I want to start workflows on my PDA.

Figure 6.7: The PDA Settings Subform of the PAVONE Sales Application

When saving the Personal Settings document, a Lotus Script library computes Notes selection formulas for each table of the DB2 Everywhere database on the PDA, based on the selection of the user. These formulas are stored in separate fields in the Personal Settings document and are used by the synchronisation tool for data selection. Figure 6.8 shows an example for a selection formula for the correspondence table of the DB2 Everywhere database on the PDA. It is based on the user-defined selection of Fred Bloggs/SYMANTEC as shown in Figure 6.7:

```

SELECT
(@Contains(@Uppercase(Form);'MEMO':'DOCIN')) &
((((@Contains(@Uppercase(Company:CompanySynonym);
'LOTUS':'IBM':'SYMANTEC')) |
(@Contains(ProductOpportunity;'Norton Antivirus')) &
(ComposedDate>=@Adjust(@Today;0;-3;0;0;0) |
LetterDate>=@Adjust(@Today;0;-3;0;0;0))) |
(@Contains(@Uppercase(wfMemberList);'*':
@Uppercase('CN=Fred Bloggs/O=SYMANTEC')) &
(wfReviewDate=' ' | wfReviewDate<=@Adjust(@Today;0;0;7;0;0))))

```

Figure 6.8: Example formula for the user-defined data selection

6.5.3 Modifications to the Process Engine

To give the user the ability to complete his tasks on the PDA, a small modification to the process engine is necessary. The process definition for structured processes is stored in the Process Definition Database. When completing a task from a structured process using the Notes client, the Espresso engine looks up the next possible tasks in this database. A process can be designed in such a way that the user has to select from a list, to which task or tasks a document is routed next. This is modelled using “Exclusive Choice” or “Multiple Choice” links. The Process Definition database is not available on the PDA for data volume reasons, which means that this kind of process cannot be used on the PDA. The workflow engine has been changed in such a way that if there are "Multiple Choice" or "Exclusive Choice" links, the possible successor tasks are stored in a field of the routed document and transferred to the PDA. This information is displayed when completing a task on the PDA, and the user can select the next task as he usually does when using the Notes client.

6.5.4 Server-Based Routing and Server-Based Validation

As mentioned in Chapter 5, server-based routing is the only choice when using PDAs. The server-based routing has been implemented using a Notes agent. This agent should run on new and modified documents on the Domino server. It tries to

route documents which have been completed on the PDA. If an error occurs and the document cannot be routed to the next task or tasks, the failed completion is mentioned in the workflow log of the document. Additionally an email containing the error message and a link to the document is sent to the user who tried to complete the task. The document comes back to their "My current tasks" list. The user can try to solve the problem and complete it again.

A similar method is used for the server-based validation which has been mentioned in Chapter 5 as the better choice for complex validations. A Notes agent running on new and modified documents on the Domino server tries to validate the documents which have been created or changed on the PDA, based on the validations developed for the Notes form. If an error occurs, the error message and a link to the document is sent to the user who created or changed the document on the PDA.

6.6 Synchronisation Setup

Extended Systems XTNDConnect 2.4 is used for synchronisation between PAVONE Sales and the IBM DB2 Everywhere database on the PDA. Refer to Chapter 4.2 for an overview of the architecture and features of XTNDConnect 2.4.

6.6.1 Synchronisation

XTND Connect uses different kinds of so-called Actions for synchronisation which are grouped in so-called Action Sets. The following types of Actions are available:

- Database Action

"Database Actions are used to synchronise a PDB on the Palm OS mobile with any ODBC data source that's accessible by the XTNDConnect Server service or with major groupware products, such as Lotus Notes." [ExtendSys 2000, p. 160]

- File Action

"File Actions are used to transfer PDBs (Pilot Databases) or PRCs (Pilot Resources) between a Palm OS mobile and the XTNDConnect Server service (or vice-versa)." [ExtendSys 2000, p. 160]

- Script Action

"Script Actions enable the system administrator to use VBScript to (amongst other things) directly program some of the underlying XTNDConnect Server

objects. This gives the administrator the ability to extend XTNDConnect Server in any way required." [ExtendSys 2000, p. 160]

The main Action Set of the prototype is called PAVONE Sales and consists of six Database Actions:

- Sales Contacts (Two way synchronisation)
- Sales Correspondence (Two way synchronisation)
- Sales Forecast (Two way synchronisation)
- Sales Settings (Server to PDA synchronisation only)
- Sales Favourite List (Server to PDA synchronisation only)
- Sales Startable Processes (Server to PDA synchronisation only)

The first three Database Actions synchronise in both directions because they synchronise DB2 Everywhere tables containing information which can be changed on the PDA by the user. The latter three Database Actions synchronise the system tables needed by the PDA front-end. Because these tables cannot be changed by the user on the PDA they are synchronised from the server to the PDA only.

Further Actionsets will be "Backup" or "Install/Upgrade PAVONE Sales", consisting of File Actions for backing up the PDA, and installing or upgrading the PDA front-end software. A detailed description of the setup of each Action can be found in Appendix B.

6.6.2 Data selection

In addition to the mapping of DB2 Everywhere columns to Lotus Domino fields, a Database Actions controls the data selection for synchronisation.

XTNDConnect has the ability of using TAGS for handling some synchronisation parameters dynamically. TAGS are variables which can be used for most of the parameters of an Action. The TAG values are read from server databases at the beginning of the synchronisation and are valid during the whole synchronisation process. In the most cases TAGS are used to provide more flexibility for the synchronisation process and to combine several similar Actions into a single one by handling some values dynamically.

An example for a TAG is the path of a user's mailfile on a Domino server. Based on the user's name it can be read from the server's Name & Addressbook and can then be used during a Database Action for finding the correct database for each user.

In the prototype, TAGS are used for setting up the user-defined data selection. Based on the user name, XTNDConnect finds the correct Personal Settings document in the PAVONE Sales Application database, reads the fields which contain the selection formulas for each DB2 Everywhere table, and stores each selection formula in a different TAG. In the data selection for the Database Action the option "custom" [cmp. ExtendSys 1999, pp. 202] is used and one of these TAGS is assigned. The whole process how the PDA interacts with XTNDConnect and the Domino server is shown in Figure 6.9.

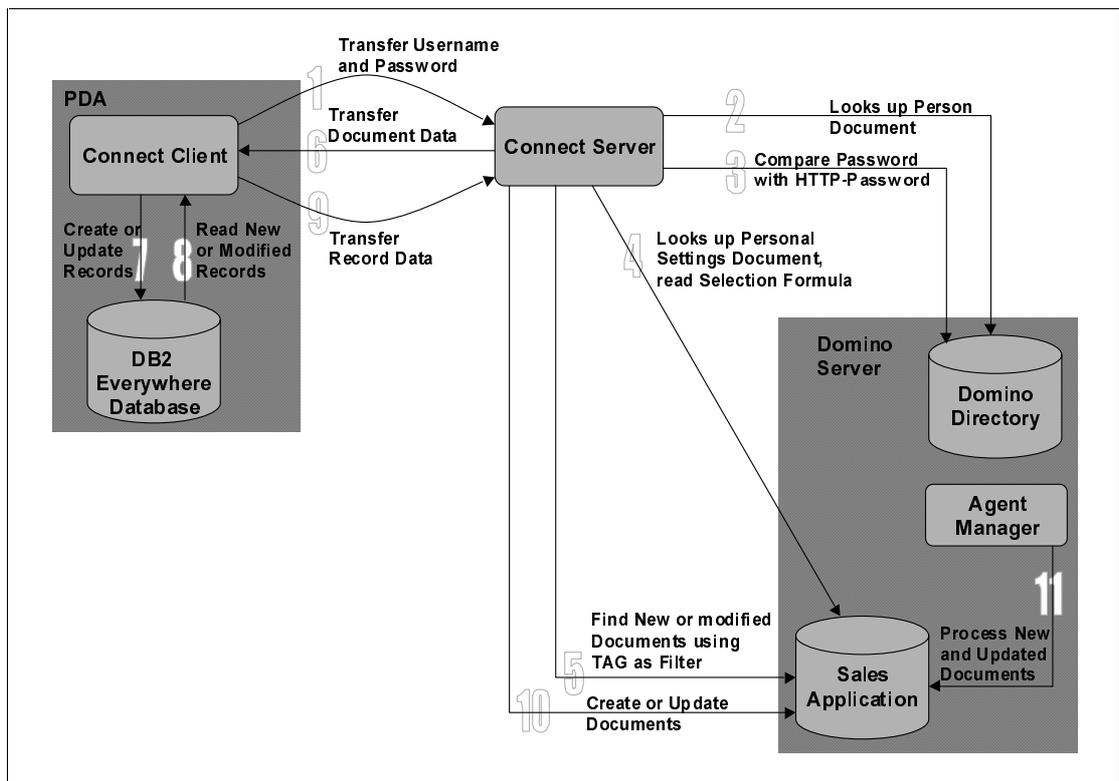


Figure 6.9: Synchronisation Process with XTNDConnect

6.7 Data Volumes on the PDA

Handling the data load has been described in Chapter 5 as a main aspect for the integration. The data load for this prototype consists of a fixed and a variable part. The fixed part consists of the memory used by the synchronisation engine, the database engine, and the front-end application. The variable part contains the records synchronised to the PDA. Table 6.1 shows the amount of memory needed by the prototype. The memory needed by the records in the tables can hardly be measured because each record type contains a field which can store up to 10000 characters

representing the rich-text field of the corresponding Notes documents. The amounts in Table 6.1 are average values taken from the operative sales system at SYMANTEC which has been used for the integration.

| Application / Record | Memory Amount |
|-------------------------------|----------------------|
| Fixed Data Loads | |
| Connect Client | 113 Kb |
| Connect Client Plugin for DB2 | 24 Kb |
| DB2 Everywhere engine | 100 Kb |
| Front-end | 78 Kb |
| Variable Data Loads | |
| Contact Record | 0.35 Kb |
| Memo Record | 1.93 Kb |
| Letter Record | 9.21 Kb |
| Document Received Record | 8.21 Kb |
| Forecast Record | 0.42 Kb |

Table 6.1: Example Data Loads of the Prototype

The total memory needed for the front-end, database engine and synchronisation engine is 315 Kb whereby it should be considered that the front-end is developed as a prototype which has not been optimised regarding the memory yet. The average sales person at SYMANTEC was responsible for roughly 100 customers. This means that at least he needs 77 Kb for the contact and forecast records, assuming that there is exactly one forecast for each customer. A Palm OS PDA is equipped with at least 2 Mb of RAM (cmp. Chapter 4.1.1). Assuming that one third of the available memory is used for the built-in applications and third-party tools, there is roughly 600 Kb available for correspondence records, e.g. 200 memos and 60 letters. Using a device equipped with 8 Mb of RAM it could be 1000 memos and 300 letters without using all of the available memory.

These are only exemplary numbers taken from a single project but they show that a today's PDA is capable of storing all sales-related information needed by a sales person on the road.

7 Conclusions and Future Perspectives

This chapter provides the conclusions regarding the use of PDAs in process-driven Enterprise sales systems. A main focus here lies on finding the business value of the solution. It also takes a look at the current research in the area of mobile computing to provide some future perspectives regarding the use of PDAs as part of the corporate IT-structure of a company.

7.1 Conclusion

As seen in the description of the solution in Chapter 6, the problems described in Chapter 3 can be addressed by using PDAs in sales environments.

The situations in which a notebook is lacking regarding mobility, as described in Chapter 3.2, can be easily handled by the PDA technology. The more mobility is needed the more effective is the use of PDAs, because of their size and their pen-based user interface, in which the concepts for user interface design described in Chapter 5.3 must be considered. Using the concepts for handling the dataload on such devices, as described in Chapter 5, shows that the memory capacity of PDAs is nowadays large enough to hold the information a sales person needs when on the road. Also the processing capacity of PDAs is sufficient when the concepts for workload management, which are explained in Chapter 5.2, are used.

The security aspects mentioned in Chapter 3.3.1 are only partly solved in the developed prototype. The technology used for the prototype on the one hand secures the corporate systems against unauthorised access if the PDA device gets lost or is stolen, and also contains a solution for encrypting information transferred between the mobile device and the enterprise system. But still a solution must be found which locks the device itself against unauthorised access. There is no product on the market at the moment which can solve this problem without the hassle of entering passwords at the start-up, but devices using the fingerprint as user authorisation are announced for the near future.

The administration efforts described in Chapter 3.3.2 for mobile devices cannot be removed completely but they can be reduced by either creating a policy for the hardware and software used and also providing a user interface which is easy to use without much training. The synchronisation engine nearly provides the potential for a complete integration of these devices by providing automatic backup and software

update features. This shows that PDAs can get much closer to the aim of being handled like a network integrated workstation.

The concept of server-based routing (cmp. Chapter 5.2.1) combined with the data selection concepts (cmp. 5.1.2, 5.1.3) make it possible to integrate PDAs into the corporate sales processes and transform them from a personal information manager into a corporate sales information manager.

7.2 Business Value

The business value of the integration can be measured by comparing the costs of deploying and running the application with the benefits the application provides. When targeting the costs for the application first the total cost of ownership (TCO) for the device itself must be regarded. The Gartner Group measures the TCO for Palm OS-based PDA as being US\$ 2693. This calculation includes the capital needed to purchase the device (24%), the costs for technical support (13%), the costs for administrating the devices (23%) and the costs for the time the user operates with the device (40%) [Dulaney 1999]. Because the cost for the user operating with the device also includes the time he spends on synchronising information, these costs mainly depend on the synchronisation engine used, and can probably be reduced using data selection concepts as described in Chapter 5.1.2.

The Gartner Group report focuses only on the use of the standard application delivered with the PDA. The integration of PDAs into Enterprise sales systems additionally involve costs for the development of the PDA application and the licenses for the synchronisation engine.

The benefits of integrating PDAs into a process-driven sales system are difficult to measure. Usually when implementing a workflow management system the benefit can be calculated by measuring the time which is saved because of the use of the system, e.g. decreased rest periods. When integrating PDAs into a process-driven system, these savings depend on how often the user synchronises the PDA with the server, because of server-based routing technologies (cmp. Chapter 5.2.1). Wireless technologies can be used to shorten the intervals between synchronisation while on the road, but are very cost intensive at the moment. Chapter 7.3 will show the future perspectives of such technologies. Additional timesavings are given because any information is directly entered into the corporate system. There is no manual intervention needed for transferring the information from the PDA into the corporate

system meaning there is no time wasted, as there would be if a paper-based organiser or a conventional PIM were used. Aside from saving time the costs for printouts from the corporate sales information system which are needed by the sales person on the road will be decreased. The exact cost savings depend on the sales processes running within the sales department and can differ between industries. In the affiliated project with SYMANTEC, the sales processes have been restructured just before the research for this thesis started. This means that the numbers needed for calculating the cost savings were not available.

As an example some fictive numbers could be taken: Assuming that a sales person needs one hour a week for printing out information from the sales information system and entering changes he made on the road into the system, and the personnel expenditure for that sales person are US\$ 100,000.00 a year. Assuming a 40-hour working week this means a cost saving of US\$ 2,500.00 per year for the company plus the saving for the material of the printouts. This calculation shows only the possible cost savings caused by the integration of PDAs into corporate sales information systems. But the real benefit of the use of PDAs is to have any sales-related information available anytime and anywhere when facing the customer. This benefit accruing from customer relationship management cannot be measured in cash terms. But by having lead-related information available during a sales conversation, no matter where and when this conversation happens, the satisfaction of the sales lead contact can be increased which means a higher probability of winning the sale. How much the probability can be increased can only be measured using long term statistics after integrating the PDAs.

The fact is that the use of PDAs in the sales environment can increase customer satisfaction and consequently increase the probability of winning a sale. Whether the revenue coming from this increased probability of winning sales can exceed the costs for integrating PDAs into corporate IT-structures is not known yet. But regarding the future perspectives of PDAs it is important to think about this topic now, so as not to fall in behind in the near future.

7.3 Future Perspectives

“Gartner Group predicts there will be at least 30 million hand-held computers worldwide by 2003, and more than 100 million by 2009, Giga (remark of the author: “GIGA Information Group”) believes the number of wireless and hand-held devices

is going to explode in the next two years, while Dataquest claims that more than 60 per cent of devices accessing the Internet by 2004 will be wireless." [Lynch 1999]

This quote demonstrates what all the major IT research organisations predict for the future at the moment, and shows the way many hardware manufacturers and software development companies are going right now. The aspects the IT industry will concentrate on in the near future are wireless mobile devices and the integration of these devices into corporate systems.

It seems to be that there is a radical change at the moment regarding PDA devices. Although, compared with the technology of desktop PCs, the devices of Palm Inc. did not change so much during the last 5 years there has been a rapid growth in sales of such devices during the last year. Additionally Microsoft announced the launch of Pocket PC, a completely redesigned version of Windows CE, for the first quarter of 2000. This operating system will provide an easy user interface like Palm OS combined with the multimedia capabilities Windows CE already has. The increasing interest in mobile devices continues in the efforts of software development companies who try to create interfaces for their sales solutions which make integration with PDAs possible.

The next step is to enhance mobile devices with wireless capabilities. At the moment the Wireless Application Protocol (WAP) seems to be the answer to the question of wireless Internet access. It enables the user of a special mobile phone or a PDA equipped with a wireless modem to view the content of Internet pages on demand in a format optimised for the small screen of such devices. This means a change from information synchronisation as described in this thesis to information request on demand everywhere and anytime. Also it allows a two-way communication with such Internet pages, e.g. for shopping purposes. Another benefit of wireless devices is that their location can be determined roughly. The opportunity to know the user's location anytime can open new markets for applications. Regarding the sales environment the following scenario would be possible:

A sales person is on a customer site and the appointment unexpectedly takes longer than planned so that it extends into lunchtime. Because the corporate sales information system is able to determine the position of the sales person it could send information about the nearest restaurant which is known to be appropriate for business lunches. Also the sales information system probably knows about special preferences of that customer regarding the type of cuisine and takes this information

into consideration when choosing the restaurant. The sales person gets this information pushed into his PDA and is able to answer with a request for a reservation of a table for that restaurant.

Although this sounds like technology of the far future it could be implemented today with integrating the WAP technology into corporate systems.

Whether WAP will be the solution or any other wireless technology cannot be predicted at the moment but it is certain that the use of wireless services will increase in the next few years. This prediction is backed up by the recent partnerships built up between the PDA manufacturers and mobile phone manufacturers, for example PSION coming together with the mobile phone manufacturer Ericsson or Palm Inc. joining forces with Nokia to develop smart phones with PDA capabilities.

The bottom line is that in the near future mobile devices will probably be used with the same casualness as PCs are used in organisations nowadays. To get the most out of this fast-growing technology companies should try to keep up-to-date in order to being able to manage their future markets.

III Bibliography

[Casio 2000]

Casio Cassiopeia Product Specifications, Casio Inc., Dover, NJ USA 07801, 2000. From: <http://www.casio.com> on 15/04/2000

[Compaq 2000]

Compaq Handhelds Product Specifications, Compaq Computer Corporation, Houston, TE USA 77269, 2000. From: <http://www.compaq.com/products/handhelds> on 15/04/2000

[Craig / Leyden 1999]

Craig, A.; Leyden J.: Palm gets smart phone deal. In: Network News 1999, No. 42 (20/10/1999), p. 4

[Dulaney 1999]

Dulaney, K.: TCO for PDAs: Higher Than Expected. Research Note from 27/07/1999, available from Gartner Group: <http://www.gartner.com>

[Encarta2000]

Microsoft Encarta Online Encyclopedia 2000. From: <http://encarta.msn.com> on 11/06/2000

[Ericsson 2000]

Ericsson MC 218 Mobile Companion Specification, Ericsson AB, Stockholm, Sweden, 2000. From: <http://www.ericsson.com> on 15/04/2000

[ExtendSys 1999]

XTNDConnect Server for Palm OS - User manual, Extended Systems Inc., Boise/Idaho, 1999

[ExtendSys 1999a]

XTND Connect Server, Mobile Synchronization and Management Solution for the Enterprise. From: <http://www.extendedsystems.com/products/serversync> on 13/12/1999

[Handspring 2000]

Handspring Visor and Visor Deluxe, Handspring Inc., Mountain View, CA USA 94043, 2000. From: <http://www.handspring.com> on 15/04/2000

[Hollingsworth 1995]

Hollingsworth, D.: Workflow Management Coalition – The Workflow Reference Model, Vers. 1.1, Workflow Management Coalition, 1995. From: <http://www.aiim.org/wfmc/standards/docs/tc003v11.pdf> on 12/06/2000

[Keil-Slawik 1999]

Keil-Slawik, R.: Lecture script “Software Ergonomics” at University of Paderborn, winter term 1998/1999. Chapter 17 from 21/01/1999

[Kennedy 1999]

Kennedy, S.: Save as houses. In: Computer Weekly 1999, No. 42 (21/10/1999), p. 46

[Loane 1999]

Loane, A.: Mobile device boom is IT management headache. In: Network News 1999, No. 49 (08/12/1999), p. 12

[Lynch 1999]

Lynch, M.: Hand-helds mean business. In: Computer Reseller News 1999, No. 48 (01/12/1999), pp. 52-54

[McCrea 1999]

McCrea, P. N.: Cost/Benefit Analysis for Technology Enabled Selling. From: http://www.c3i-inc.com/wp_cost_ben.htm on 10/12/1999

[Muench 2000]

Muench, C.: Choosing the Right Programming Language. From: <http://www.microsoft.com/pocketpc/columns/vbvsvc.asp> on 17/05/2000.

[Murphy 1999]

Murphy, I.: Windows CE does the fuel monty. In Computing 1999, No. 45 (18/11/1999), p. 48

[Nastansky et al. 1995]

Nastansky L.; Schicker, T; Behrens, O. M.; Ehlers, P.:
Büroinformationssysteme. In: Dangelmaier, W.; Herold, W.; Fischer, J.;
Nastansky, L.; Wolff, R.: Bausteine der Wirtschaftsinformatik, S+W Steuer-
und Wirtschaftsverlag, Hamburg 1995, pp. 267-371

[Nastansky / Hilpert 1994]

Nastansky, L.; Hilpert, W.: The GroupFlow System: A Scalable Approach to Workflow Management between Cooperation and Automation, February 1994

[Palm 1999]

Palm OS Programmer's Companion, Palm Inc., Santa Clara, CA USA 95052, 1999

[Palm 2000]

Palm Inc. Product Family, Palm Inc., Santa Clara, CA USA 95052, 2000.
From: <http://www.palm.com> on 15/04/2000

[PAVONE 1999]

PAVONE First Steps Espresso, PAVONE Informationssysteme GmbH, Paderborn, 1999

[PCTechGuide 2000]

The PC Technology Guide: PDAs. From:
<http://www.pctechguide.com/25mob3.htm> on 11/06/2000

[Philips 2000]

Philips Nino Product Specifications, Philips Electronics N.V., Groningen, Netherlands, 2000. From: <http://www.nino.philips.com> on 15/04/2000

[PSION 2000]

Psion Handheld Specifications, Psion Inc., Concord, MA USA 01742, 2000.
From: <http://www.pSION.com> on 15/04/2000.

[Shiang / Hodges 1999]

Shiang, D. A.; Hodges, J.: Sales Force Automation Software: 1999 Worldwide Markets and Trends, October 1999. Available at IDC Net:
<http://www.idc.net>, report no. W20345

[TRG 2000]

TRGpro Handheld Computer, TRG Products Inc., Des Moines, IA USA 50322, 2000. From: <http://www.trgpro.com> on 15/04/2000

[Webopedia 2000]

Ecommerce Webopedia: PIM. From: <http://e-comm.webopedia.com/Term/P/PIM.html> on 11/06/2000

[WfMC 1999]

Workflow Management Coalition: Terminology & Glossary, Issue 3.0, February 1999. From: <http://www.aiim.org/wfmc/standards/docs/glossy3.pdf> on 12/06/2000

IV Table of Figures

| | |
|--|----|
| Figure 2.1: Screenshot of a typical PC-based PIM (Lotus Organizer 6.0) | 6 |
| Figure 2.2: Workflow System Characteristics [Hollingsworth 1995] | 9 |
| Figure 2.3: Workflow Reference Model – Components & Interfaces [Hollingsworth 1995, p.20] | 10 |
| Figure 2.4: The Sales Process [McCrea 1999] | 13 |
| Figure 3.1: Mobility Aspects of some exemplary Places | 16 |
| Figure 4.1: Example PDA devices of the dominant PDA operating systems | 21 |
| Figure 4.2: XTNDConnect Architecture [ExtendSys 1999a] | 34 |
| Figure 4.3: Interfaces of PDA Application Modules | 35 |
| Figure 5.1: Three Step Data Volume Reduction | 39 |
| Figure 5.2: Client-based Routing vs Server-based Routing | 43 |
| Figure 5.3: Server-based Routing for PDAs | 44 |
| Figure 5.4: Server-based Validation on PDAs | 45 |
| Figure 6.1: Architecture of the Sales Application Database | 48 |
| Figure 6.2: The PAVONE Sales Architecture [PAVONE 1999, p. 3] | 49 |
| Figure 6.3: Architecture of the Prototype | 50 |
| Figure 6.4: DB2 Everywhere Data Structure | 51 |
| Figure 6.5: PDA Front-end – Form Structure | 52 |
| Figure 6.6: PDA Front-end – different Form Types | 53 |
| Figure 6.7: The PDA Settings Subform of the PAVONE Sales Application | 55 |
| Figure 6.8: Example formula for the user-defined data selection | 56 |
| Figure 6.9: Synchronisation Process with XTNDConnect | 59 |

V Tables of Tables

Table 3.1: Notebook Dimensions and Battery Life of sample Manufacturers17

Table 4.1: Specifications of sample Palm OS-based PDAs22

Table 4.2: Specifications of sample Windows CE-based PDAs23

Table 4.3: Specifications of sample EPOC-based PDAs.....25

Table 6.1: Example Data Loads of the Prototype.....60

VI Table of Abbreviations

| | |
|--------|--|
| API | Application Programming Interface |
| CDK | Conduit Development Kit |
| CRM | Customer Relationship Management |
| DB | Database |
| FSTN | Flat SuperTwist Nematic |
| GCC | Gnu Compiler Collection |
| IrDA | Infrared Data Association |
| Kb | Kilobyte |
| LCD | Liquid Crystal Display |
| LAN | Local Area Network |
| Mb | Megabyte |
| OS | Operating System |
| PC | Personal Computer |
| PDA | Personal Digital Assistant |
| PDB | Pilot Database |
| PIM | Personal Information Manager |
| PRC | Pilot Resource |
| RAD | Rapid Application Development |
| RAM | Random Access Memory |
| RDBMS | Relational Database Management System |
| ROM | Read Only Memory |
| SDK | Software Development Kit |
| SFA | Sales Force Automation |
| SQL | Structured Query Language |
| TCO | Total Cost of Ownership |
| TCP/IP | Transmission Control Protocol / Internet Protocol |
| TFT | Thin Film Transistor |
| WAN | Wide Area Network |
| WAP | Wireless Application Protocol |
| WAPI | Workflow Application Programming Interface |
| WfMC | Workflow Management Coalition |

VII Eidesstattliche Erklärung

Ich erkläre hiermit an Eides Statt, daß ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Paderborn, 25.07.2000

.....
Valentin Balzer

VIII Appendix

Appendix A: Prototype Installation & Configuration

Appendix B: Prototype Technical Documentation

Appendix A: Prototype Installation & Configuration

- 1 Contents of the CD-ROM
- 2 General Information
- 3 System Requirements
- 4 Installation
- 5 Configuration

1 Contents of the CD-ROM

| Directory | Files |
|----------------------|--|
| Documentation | FirstSteps.pdf Install.pdf |
| Installation | Readme.txt |
| PAVONE_Sales | |
| Databases | Es_mobile.nsf Sf_org.nsf Sf_proc.nsf Sf_set.nsf |
| Organization_Modeler | Setup.exe Additional files needed by the setup program |
| Process_Modeler | Setup.exe Additional files needed by the setup program |
| XTNDConnect | Connect24eval.exe |
| DB2_Everywhere | DB2Everywhere.exe |
| Frontend_Source | PAVONE Sales (Project Directory) |
| XTNDConnect_Config | Pavone_Sales.acf |
| Presentation | Diploma_Thesis_Valentin_Balzer.ppt |
| Thesis | Diploma_Thesis_Valentin_Balzer.doc Diploma_Thesis_Valentin_Balzer.pdf |

2 General Information

Any software on the CD-ROM is the property of the appropriate manufacturer and is copyrighted. Any reproduction in whole or in part is strictly prohibited. The software on the CD-ROM is provided only for demonstration purposes and must not be used operationally. Neither the manufacturer of the software nor the author of this document is liable for any possible damage arising due to the installation and use of the software. The author recommends the installation of the prototype only on test environments. The installation of the different modules should be carried out only by a person with expert knowledge in network computing, especially in Microsoft Windows NT and Lotus Domino. When installing and running the software please always have in mind that you are using a prototype, not a released product. When

facing problems during the installation or configuration of the prototype please refer to the electronic documentation delivered with each of the modules.

3 System Requirements

To run Extended Systems XTNDConnect Server a PC running the operating system Microsoft Windows NT 4.0 Workstation or Server with at least 20 MB of free hard disk space and an installed TCP/IP-protocol with a fixed IP-address is required. To connect to the PAVONE Sales system a Lotus Domino Server Release 4.6 or 5 is needed. It is recommended to install the Lotus Domino Server on the same machine as the XTNDConnect Server, but it is also possible to install it on a separate machine which also needs a running TCP/IP-protocol and a fixed IP address. The machine running Lotus Domino Server needs an additional 50 MB of free hard disk space. A PDA running Palm OS version 3.3 or higher with at least 500 Kb of free memory is required for the PDA software installation.

4 Installation

4.1 PAVONE Sales Databases

Shut down the Lotus Domino Server. Copy all database files from the directory *Install\PAVONE_Sales\Databases* of the CD-ROM to a directory below the data-directory of the Lotus Domino Server. Use the Windows Explorer to remove the read-only flags from the database files on the hard disk. Restart the Lotus Domino Server.

4.2 PAVONE Organization Modeler

This application is optional and not absolutely necessary for running the prototype. For installation run the file *setup.exe* from the directory *Install\PAVONE_Sales\Organization_Modeler* of the CD-ROM. Follow the instructions given during the installation.

4.3 PAVONE Process Modeler

This application is optional and not absolutely necessary for running the prototype. For installation run the file *setup.exe* from the directory *Install\PAVONE_Sales\Process_Modeler* of the CD-ROM. Follow the instructions given during the installation.

4.4 XTND Connect Server

On the Microsoft Windows NT 4.0 machine run the file *connect24eval.exe* from the directory *Install\XTNDConnect* of the CD-ROM. Follow the instructions given during the installation. Install XTNDConnect Server to a directory of your choice (hereinafter referred to as *XTNDConnect-Directory*). After the installation do not start the XTNDConnect Server directly. Copy the file *PAVONE_Sales.acf* from the directory *Install\XTNDConnect_Config* of the CD-ROM to the directory *XTNDConnect-Directory*. Use the Windows Explorer to remove the read-only flag from the file *PAVONE_Sales.acf* on the hard disk.

4.5 IBM DB2 Everywhere

Run the file *DB2Everywhere.exe* from the directory *Install\DB2_Everywhere* on the CD-ROM. Install IBM DB2 Everywhere to a directory of your choice (hereinafter referred to as *DB2-Directory*). Follow the instructions given during the installation.

4.6 Metrowerks Codewarrior 6

This application is needed for viewing and changing the source-code of the PDA front-end. It is not necessary for running the prototype.

For installation run the file *cw_lite_palm_win.exe* from the directory *Install\Codewarrior* of the CD-ROM. Follow the instructions given during the installation.

4.7 Source-Code of the PDA Front-end

Copy the directory *Install\Frontend_Source* from the CD-ROM including all files and sub-directories to a directory of your choice on the hard disk (hereinafter referred to as *Source-Code-Directory*). Use the Windows Explorer to remove the read-only flag from all copied files on the hard disk.

4.8 PDA Applications

Use the installation tool delivered with the Palm OS PDA to install the following files to the PDA:

| Filename | Path | Comment |
|---------------------------------|---|--|
| PAVONE Sales.prc | <i>Source-Code-Directory</i> | Front-end application of the PDA |
| ConPilot.prc ConnectDB2e.prc | <i>XTNDConnect-Directory\Clients\PalmOS</i> | XTNDConnect client and plugin for DB2 Everywhere |
| DB2eCmp.prc DB2eRunTime.prc | <i>DB2-Directory\PalmOS</i> | Runtime engine of DB2 Everywhere |
| qbe.prc | <i>DB2-Directory\PalmOS\Samples\QBE</i> | Table viewer for DB2 Everywhere |

5 Configuration

5.1 PAVONE Sales

Configure the connections between the Sales Application database, the Sales Organization database, the Sales Settings database and the Sales Process Definition database as described in the file *Install.pdf* in the directory *Documentation* on the CD-ROM.

An administration-ID is needed which will be used for the XTNDConnect Server to access the databases. Create this ID using the Lotus Notes Administration tool. Add the administration-ID to the ACL of each of the Sales databases. Grant the ID at least Editor-access and select the role “Administrator”.

Sign the design elements of the Sales Application database with the administration-ID. Grant the administration-ID the rights to run any kind of agent on the server. Use the administration-ID to enable the agent “Process PDA documents” in the Sales Application database.

Create a user-ID for the PDA-user using the Lotus Notes Administration tool or use an existing ID. Add this ID to the ACL of each of the Sales databases. Grant the ID at least Author-access and select the role “Everybody”. Assign an HTTP-password for that user in the Domino Directory.

Create a Personal Settings document for the user and fill out the Favourites list and the section “PDA Settings”. Use the Personal Settings document of “Arno Rautmann/TECONE” as template.

5.2 XTNDConnect Server

Start the *XTNDConnect Server Administration* application. Open the configuration file *PAVONE_Sales.acf* which resides at *XTNDConnect-Directory*.

Setup the security settings as described in the file *Connect Manual (palmOS).doc* in the directory *XTNDConnect-Directory\Docs*, Chapter “The service security settings”.

Setup the service settings for Palm OS as described in the file *Connect Manual (palmOS).doc* in the directory *XTNDConnect-Directory\Docs*, Chapter “The XTNDConnect Server service - further configuration options”. In the service settings dialogue define the IP-address and port of the XTNDConnect Server and define the “Palm OS Config File Path” to point to the file *PAVONE_Sales.acf* in the directory *XTNDConnect-Directory*.

Configure the connection to your Lotus Domino server with the following steps:

1. In the XTNDConnect Server Administration application, double click on the root node: “Connect Configuration”. The “System Settings” form will appear.
2. Select the “Authentication” tab.
3. Double click on the entry: “[Lotus Notes] Authentication Service”. The “Notes Authentication Settings” form will appear.
4. Select the “General” tab.
5. In the field “ID File” enter the path and filename of the administration-ID you created when configuring the PAVONE Sales databases.
6. In the field “Password” enter the password of the administration-ID.
7. Select the “Tags” tab.
8. Double click each of the Tags one after the other to open the window “Lotus Notes Authentication Plugin Tags”
9. For every Tag change the field “Notes Server” from “Server/TECONE” to the full hierarchical name of your Lotus Domino server.
10. For the Tags “SALES_CONTACT_SELECT”, “SALES_CORRESPONDENCE_SELECT” and “SALES_FORECAST_SELECT” change the field “Notes Database” to the path and filename of the PAVONE Sales Application database on your Lotus Domino server.

The Actions of the Action Set “PAVONE Sales” must be adapted to your Lotus Domino Server and the installed PAVONE Sales databases. Refer to the file *Connect*

Manual (palmOS).doc in the directory *XTNDConnect-Directory\Docs*, Chapter "Database Actions" for how to change database actions.

For every Action change the field "Server Name" in the tab "General" of the form "Database Action Properties" from "Server/TECONE" to the full hierarchical name of your Lotus Domino server.

For the Actions "Sales Contacts", "Sales Correspondence", "Sales Forecast", "Sales Favourite List", "Sales Startable Tasks" change the field "Database Name" to the path and filename of the PAVONE Sales Application database on your Lotus Domino server.

For the Action "Sales Settings" change the field "Database Name" to the path and filename of the PAVONE Sales Settings database on your Lotus Domino server.

Restart the XTNDConnect Server service.

5.3 XTNDConnect Client

Connect your Palm cradle to a serial port of a machine which is network connected via TCP/IP to the machine running the XTNDConnect Server service. Run the file "ConnectProxy.exe" which resides at *XTNDConnect-Directory\Proxy*. You can either run the application from the network drive, or copy the file to your local machine and run it from there.

On your PDA start the "Connect" application. Open the menu "Server..." and type in the IP-address and port of the machine running the XTNDConnect Server. Tick the checkbox "Use XTNDConnect Proxy". Close this menu using the "OK" button and open the menu "Identity...". In the field "User name" enter the full hierarchical name of the user-ID you created when configuring the PAVONE Sales databases. Enter the HTTP-password of that user in the field "Password". Tick the checkbox "Remember password".

5.4 Running the PDA Application for the First Time

On your PDA start the "Connect" application. Press the button "Connect". The PDA should now connect to the XTNDConnect Server service. You will be asked for the public key of the XTNDConnect Server. You can find the key in the security settings of the XTNDConnect Server Administration application. Enter the key and re-connect to the server. After a successful connection you will find several Action Sets displayed on your screen. Select the Action Set "PAVONE Sales". Connect to the server again using the button "Connect". Based on your Personal Settings document

in the PAVONE Sales Application database the desired documents will now be transferred to your PDA. The DB2 Everywhere tables will be created on your PDA automatically during the synchronisation. Depending on the server performance the synchronisation can take several minutes.

Because of a bug in the XTNDConnect plugin for DB2 Everywhere the following procedure is needed:

1. On your PDA start the “QBE” application.
2. In the top right corner tap the label “Tap here to pick a table”.
3. From the appearing list select the first entry.
4. After a few seconds a “Fatal Exception” will arise! You have to reset your PDA by using the displayed button or the switch on the back of your PDA.

Repeat the procedure for each of the entries in the list of the “QBE” application. The whole procedure is needed only once after the automatic creation of the tables on your PDA. Use the “Connect” application to synchronise with the server again. Now you can start using the PAVONE Sales front-end. The “QBE” application is not needed any more on the PDA.

Appendix B: Prototype Technical Documentation

- 1 Enhancements of PAVONE Sales
- 2 Setup of Extended Systems XTNDConnect Server

1 Enhancements to PAVONE Sales

1.1 Subforms

1.1.1 \$PDASync

This Subform contains all additional fields needed for synchronising a document with the PDA front-end. It must be inserted into any form which is used for creating documents being used on the PDA.

1.1.2 \$PDASettings

This subform contains the data selection profile. It must be inserted into the "Personal Settings" form.

1.2 Agents

Process PDA documents

This agent has two functions:

The agent updates documents which have been created or modified on the PDA.

It adds the features to the process engine which are needed for server-based routing.

The agent should be scheduled to run on all new and modified documents at least hourly on the server.

1.3 Script Libraries

1.3.1 wfDocument

The existing script library wfDocument must be enhanced to implement server-based routing. The following enhancements are needed:

Class PavoneWorkflowDocument

```
Private Sub TaskSetItemsDef(nd As NotesDocument, ptd As PavoneTaskDefinition)
```

```
...
```

```
Call nd.ReplaceItemValue("wfObject", ptd.Forms)
```

```
' PDA Synchronisation -> inserting the possible next or and xor branches
```

```
    Dim choices As Variant
```

```
    Dim str_choices As String
```

```
    choices=ptdd.GetLinkChoicesByType("Xor")
```

```
    str_choices=""
```

```
    Forall ch In choices
```

```
        str_choices=str_choices & ch & "<$%&%$>" 'PDA spezifisches Format
```

```

End Forall
str_choices=Left(str_choices,Len(str_choices)-7)
If str_choices<>"" Then
    Call nd.ReplaceItemValue("PDA_wfNextXORTasks",str_choices)

choices=ptdd.GetLinkChoicesByType("Or")
str_choices=""
Forall ch In choices
    str_choices=str_choices & ch & "<$%&%%$>" 'PDA spezifisches Format
End Forall
str_choices=Left(str_choices,Len(str_choices)-7)
If str_choices<>"" Then
    Call nd.ReplaceItemValue("PDA_wfNextORTasks",str_choices)

```

```
' *** End of PDA Code
```

```
If ptdd.HasManager Then
```

```
...
```

```
Private Sub TaskSetItemsTransition(intCompType As Integer)
```

```
...
```

```
Call .RemoveItem("wfDuration")
```

```
' PDA Synchronisation-> inserting the possible next or and xor branches
```

```
Call .RemoveItem("PDA_wfNextXORTasks")
```

```
Call .RemoveItem("PDA_wfNextORTasks")
```

```
' *** End of PDA Code
```

```
End Select
```

```
...
```

After the change the script library wfUIView must be recompiled!

1.3.2 PDASyncLib

The library PDASyncLib contains all functions needed to compute valid selection formulas from the profile values defined in the subform \$PDASettings. The following functions are included:

GetContactSelection

Computes the selection formula for the PDA-table containing the contact records.

The function is called in the QuerySave-event of the \$PDASettings subform.

GetCorrespondenceSelection

Computes the selection formula for the PDA-table containing the correspondence records. The function is called in the QuerySave-event of the \$PDASettings subform.

GetForecastSelection

Computes the selection formula for the PDA-table containing the forecast records. The function is called in the QuerySave-event of the \$PDASettings subform.

GetUsername

Extracts the hierarchical Notes username from the Personal Settings document.

ListItemToString

Converts a Notes item into a text string whereby the values in the list are separated by a colon.

ConvertQuotations

Replaces all quotation marks (") in a text string with apostrophes ('). This must be done for any selection formula which shall be used by XTNDConnect.

DateRangeKeyToSelectString

Converts a date value keyword selected in the \$PDASettings subform into a component of a selection formula.

GetCompanyPartSelection

Generates the component of a selection formula containing the selected companies.

ComputeForecastValues

Converts the forecast range of the \$PDASettings subform into a component of a selection formula.

GetWorkflowPartSelection

Generates the component of a selection formula related to the current task list and the review list.

GetWorkflowStartTasks

Determines the processes a user is allowed to start and converts them into a format usable by the PDA front-end.

1.4 PDA-related Fields

These following fields are added to a Notes document that has been changed on the PDA. They are analysed by the agent “Process PDA Documents” and removed if the agent has run successfully.

PDA_UpdatedBy

The hierarchical Notes username of the PDA user who changed a document.

PDA_wfState

Contains an “N” for an process initialised on the PDA, an “A” for an ad-hoc task initialised on the PDA and a “C” for a task which has been completed on the PDA.

PDA_wfNewType

Contains the priority of a process initialised on the PDA.

PDA_wfNewTask

Contains the name of an ad-hoc task initialised on the PDA or the name of a process initialised on the PDA.

PDA_wfNewMember

Contains the memberlist of an ad-hoc task initialised on the PDA.

PDA_wfNewTitle

Contains the title of a process initialised on the PDA.

PDA_wfNewNotification

Captures whether a process initialised on the PDA should use the mail notification feature of the workflow engine.

2 Setup of ExtendedSystems XTNDConnect Server

2.1 TAGS

TAGS are values which can be read from a Notes database at the beginning of a synchronisation. For the prototype they are used to read the selection formulas from the Personal Settings documents.

| | | | |
|-----------------|---|-----------------------------|-----------------------|
| TAG Name | SALES_CONTACT_SELECT | SALES_CORRESPONDENCE_SELECT | SALES_FORECAST_SELECT |
| Lookup Value | %NOTESAUTH_NAME% | | |
| Notes Server | Full hierarchical name of your Domino server | | |
| Notes Database | Path and file name of the PAVONE Sales Application database | | |
| Notes View Name | \$PavoneSettingPersonalView | | |
| Return Value | PDASyncContactSelect | PDASyncCorrespondenceSelect | PDASyncForecastSelect |

1.2 Database Actions

Six Database Actions are needed to run all features of the prototype. The setup of each Action is described in this chapter. Additional Actions for installing, upgrading, and backing up the software on the PDA should be designed. Refer to the file *Connect Manual (palmOS).doc* in the directory *XTNDConnect-Directory\Docs* for the setup of these Actions.

Sales Contacts

| | | | | | |
|-----------------|---|-------------|---------------|--------------------|--------|
| Action | Two Way Synchronise | | | | |
| Server Type | Lotus Domino/Notes | | | | |
| Database Type | Notes Database | | | | |
| Server Name | Full hierarchical name of your Domino server | | | | |
| Database Name | Path and file name of the PAVONE Sales Application database | | | | |
| Form Name | Address | | | | |
| Server Changes | Inserts, Updates | | | | |
| Plugin Name | DB2e Plugin | | | | |
| Table Name | ES_Address | | | | |
| Mobile Changes | Inserts, Updates Deletes | | | | |
| Filter (Custom) | %SALES_CONTACT_SELECT% | | | | |
| Columns | Server Column | Server Type | Mobile Column | Internal Data Type | Others |
| | Company | String | COMPANY | String | |
| | CompanySynonym | String | COMPANYSYN | String | |
| | Division | String | DEPARTMENT | String | |
| | CompanyOption | String | COMPANYADD | String | |
| | Lastname | String | LASTNAME | String | |
| | Firstname | String | FIRSTNAME | String | |

| | | | | | |
|------------------------|------------------------------|------------|--------------------|--------------------------|-------------|
| Columns (continued) | Salutation | String | SALUTATION | String | |
| | Titel | String | TITLE | String | |
| | Position | String | POSITION | String | |
| | PID | String | ID | String | |
| | Floor | String | BUILDING | String | |
| | Street | String | STREET | String | |
| | City | String | CITY | String | |
| | State | String | STATE | String | |
| | Country | String | COUNTRY | String | |
| | ZIP | String | ZIPCODE | String | |
| | POB | String | POBOX | String | |
| | ZIP_POB | String | ZIPCODEPOBO X | String | |
| | ZIP_GK | String | ZIPCODECOMP ANY | String | |
| | Phonenumber | String | PHONENUMBE R | String | |
| | Faxnumber | String | FAXNUMBER | String | |
| | Mobilenummer | String | MOBILENUMBE R | String | |
| | Telexnumber | String | OTHERNUMBE R | String | |
| | Email | String | EMAIL | String | |
| | WWW | String | WWW | String | |
| | Comments | RichText | COMMENTS | String | |
| | Form | String | FORM | String | |
| | AccessReader | String | ACCESSREAD ER | String | |
| | AccessAuthor | String | ACCESSAUTH OR | String | |
| | DocID | String | CONTACTID | String | Primary Key |
| | PDA_UpdatedBy | String | UPDATEDBY | String | |
| | PDA_CreateDate | Date/Time | CREATEDATE | Date/Time | Res: Day |
| PDA_CreateTime | Date/Time | CREATETIME | Date/Time | Res:Second, Time only | |
| | WfTask | String | WFTASK | String | |
| | WfReviewDate | Date/Time | WFREVIEWDAT E | Date/Time | Res: Day |
| | WfReviewDateNotifica tion | String | WFREVIEWNO TIFY | String | |
| | PDA_wfState | String | WFSTATE | String | |
| | PDA_wfNewType | String | WFNEWTTYPE | String | |
| | PDA_wfNewTask | String | WFNEWTASK | String | |
| | PDA_wfNewMember | String | WFNEWMEMB ER | String | |
| | PDA_wfNewTitle | String | WFNEWTITLE | String | |
| | PDA_wfNewNotificati on | String | WFNEWNOTIF Y | String | |
| | PDA_wfNextXOrTask s | String | WFXORBRANC HES | String | |

| Columns (continued) | PDA_wfNextOrTasks | String | WFORBRANCH ES | String | |
|------------------------|-------------------|----------------|------------------|------------------|-----|
| DB2e Info | Column | Name | Data Type | Length/ Scale | Key |
| | 0 | CREATEDATE | Date | 4/2 | |
| | 1 | CREATETIME | Time | 4/2 | |
| | 2 | COMPANY | VarChar | 50/2 | |
| | 3 | COMPANYSYN | VarChar | 50/2 | |
| | 4 | DEPARTMENT | VarChar | 30/2 | |
| | 5 | COMPANYADD | VarChar | 30/2 | |
| | 6 | LASTNAME | VarChar | 30/2 | |
| | 7 | FIRSTNAME | VarChar | 30/2 | |
| | 8 | SALUTATION | VarChar | 10/2 | |
| | 9 | TITLE | VarChar | 10/2 | |
| | 10 | POSITION | VarChar | 30/2 | |
| | 11 | ID | VarChar | 20/2 | |
| | 12 | BUILDING | VarChar | 30/2 | |
| | 13 | STREET | VarChar | 100/2 | |
| | 14 | CITY | VarChar | 50/2 | |
| | 15 | STATE | VarChar | 50/2 | |
| | 16 | ZIPCODE | VarChar | 10/2 | |
| | 17 | POBOX | VarChar | 10/2 | |
| | 18 | ZIPCODEPOBOX | VarChar | 10/2 | |
| | 19 | ZIPCODECOMPANY | VarChar | 10/2 | |
| | 20 | PHONENUMBER | VarChar | 30/2 | |
| | 21 | FAXNUMBER | VarChar | 30/2 | |
| | 22 | MOBILENUMBER | VarChar | 30/2 | |
| | 23 | OTHERNUMBER | VarChar | 30/2 | |
| | 24 | COMMENTS | VarChar | 2000/2 | |
| | 25 | ACCESSREADER | VarChar | 250/2 | |
| | 26 | ACCESSAUTHOR | VarChar | 250/2 | |
| | 27 | FORM | VarChar | 20/2 | |
| | 28 | CONTACTID | VarChar | 33/2 | Yes |
| | 29 | EMAIL | VarChar | 50/2 | |
| | 30 | WWW | VarChar | 50/2 | |
| | 31 | COUNTRY | VarChar | 50/2 | |
| | 32 | WFTASK | VarChar | 50/2 | |
| | 33 | WFSTATE | Character | 4/2 | |
| | 34 | WFREVIEWDATE | Date | 4/2 | |
| | 35 | WFREVIEWNOTIFY | Character | 1/2 | |
| | 36 | WFNEWTTYPE | Character | 1/2 | |
| | 37 | WFNEWTASK | VarChar | 1000/2 | |
| | 38 | WFNEWMEMBER | VarChar | 200/2 | |
| | 39 | WFNEWTITLE | VarChar | 50/2 | |
| | 40 | WFNEWNOTIFY | Character | 1/2 | |
| | 41 | WFXORBRANCHES | VarChar | 2000/2 | |
| | 42 | WFORBRANCHES | VarChar | 2000/2 | |
| | 43 | UPDATEDBY | VarChar | 50/2 | |

Sales Correspondence

| | | | | | |
|--------------------------|---|----------------|----------------|--------------------|-----------------------|
| Action | Two Way Synchronise | | | | |
| Server Type | Lotus Domino/Notes | | | | |
| Database Type | Notes Database | | | | |
| Server Name | Full hierarchical name of your Domino server | | | | |
| Database Name | Path and file name of the PAVONE Sales Application database | | | | |
| Form Name | Memo | | | | |
| Server Changes | Inserts, Updates | | | | |
| Plugin Name | DB2e Plugin | | | | |
| Table Name | ES_Corresp | | | | |
| Mobile Changes | Inserts, Updates Deletes | | | | |
| Filter (Custom) | %SALES_CORRESPONDENCE_SELECT% | | | | |
| Columns | Server Column | Server Type | Mobile Column | Internal Data Type | Others |
| | Company | String | COMPANY | String | |
| | CompanySynonym | String | COMPANYSYN | String | |
| | Lastname | String | LASTNAME | String | |
| | Firstname | String | FIRSTNAME | String | |
| | SendTo | String | SENDTO | String | |
| | CopyTo | String | COPYTO | String | |
| | PartnerInt | String | PARTNERINT | String | |
| | PartnerExt | String | PARTNEREXT | String | |
| | From | String | NOTEDBY | String | |
| | ComposedDate | Date/Time | COMPOSEDDATE | String | Res: Day |
| | PostedDate | Date/Time | DATEORIGINAL | String | Res: Day |
| | PostedInitials | String | AUTHORINITIALS | String | |
| | Original | String | SOURCE | String | |
| | IncludedItems | String | INCLUDEDITEMS | String | |
| | Subject | String | SUBJECT | String | |
| | Body | String | BODY | String | |
| | AddressUNID | String | CONTACTID | String | |
| | Comments | RichText | COMMENTS | String | |
| | Form | String | FORM | String | |
| | AccessReader | String | ACCESSREADER | String | |
| | AccessAuthor | String | ACCESSAUTHOR | String | |
| | DocID | String | CONTACTID | String | Primary Key |
| | PDA_UpdatedBy | String | UPDATEDBY | String | |
| | PDA_CreateDate | Date/Time | CREATEDATE | Date/Time | Res: Day |
| | PDA_CreateTime | Date/Time | CREATETIME | Date/Time | Res:Second, Time only |
| | WfTask | String | WFTASK | String | |
| | WfReviewDate | Date/Time | WFREVIEWDATE | Date/Time | Res: Day |
| WfReviewDateNotification | String | WFREVIEWNOTIFY | String | | |
| PDA_wfState | String | WFSTATE | String | | |

| | | | | | |
|------------------------|-----------------------|----------------|---------------|------------------|-----|
| Columns (continued) | PDA_wfNewType | String | WFNEWTTYPE | String | |
| | PDA_wfNewTask | String | WFNEWTASK | String | |
| | PDA_wfNewMember | String | WFNEWMEMBER | String | |
| | PDA_wfNewTitle | String | WFNEWTITLE | String | |
| | PDA_wfNewNotification | String | WFNEWNOTIFY | String | |
| | PDA_wfNextXOrTasks | String | WFXORBRANCHES | String | |
| | PDA_wfNextOrTasks | String | WFORBRANCHE S | String | |
| DB2e Info | Column | Name | Data Type | Length/ Scale | Key |
| | 0 | CREATEDATE | Date | 4/2 | |
| | 1 | CREATETIME | Time | 4/2 | |
| | 2 | COMPANY | VarChar | 50/2 | |
| | 3 | COMPANYSYN | VarChar | 50/2 | |
| | 4 | LASTNAME | VarChar | 30/2 | |
| | 5 | FIRSTNAME | VarChar | 30/2 | |
| | 6 | SENDTO | VarChar | 50/2 | |
| | 7 | COPYTO | VarChar | 50/2 | |
| | 8 | PARTNERINT | VarChar | 50/2 | |
| | 9 | PARTNEREXT | VarChar | 50/2 | |
| | 10 | NOTEDBY | VarChar | 50/2 | |
| | 11 | COMPOSEDDATE | Date | 4/2 | |
| | 12 | DATEORIGINAL | Date | 4/2 | |
| | 13 | SOURCE | VarChar | 30/2 | |
| | 14 | AUTHORINITIALS | VarChar | 5/2 | |
| | 15 | INCLUDEDITEMS | VarChar | 100/2 | |
| | 16 | CONTACTID | VarChar | 33/2 | |
| | 17 | DOCUMENTID | VarChar | 33/2 | Yes |
| | 18 | SUBJECT | VarChar | 100/2 | |
| | 19 | BODY | VarChar | 10000/2 | |
| | 20 | COMMENTS | VarChar | 2000/2 | |
| | 21 | FORM | VarChar | 20/2 | |
| | 22 | ACCESSREADER | VarChar | 250/2 | |
| | 23 | ACCESSAUTHOR | VarChar | 250/2 | |
| | 24 | WFTASK | VarChar | 50/2 | |
| | 25 | WFSTATE | Character | 4/2 | |
| | 26 | WFREVIEWDATE | Date | 4/2 | |
| | 27 | WFREVIEWNOTIFY | Character | 1/2 | |
| | 28 | WFNEWTTYPE | Character | 1/2 | |
| | 29 | WFNEWTASK | VarChar | 1000/2 | |
| | 30 | WFNEWMEMBER | VarChar | 200/2 | |
| | 31 | WFNEWTITLE | VarChar | 50/2 | |
| | 32 | WFNEWNOTIFY | Character | 1/2 | |
| 33 | WFXORBRANCHE S | VarChar | 2000/2 | | |

| | | | | | |
|--------------------------|----|--------------|---------|--------|--|
| DB2e Info (continued) | 34 | WFORBRANCHES | VarChar | 2000/2 | |
| | 35 | UPDATEDBY | VarChar | 50/2 | |

Sales Forecast

| | | | | | |
|-----------------|---|-------------|----------------|--------------------|----------|
| Action | Two Way Synchronise | | | | |
| Server Type | Lotus Domino/Notes | | | | |
| Database Type | Notes Database | | | | |
| Server Name | Full hierarchical name of your Domino server | | | | |
| Database Name | Path and file name of the PAVONE Sales Application database | | | | |
| Form Name | Forecast | | | | |
| Server Changes | Inserts, Updates | | | | |
| Plugin Name | DB2e Plugin | | | | |
| Table Name | ES_Forecast | | | | |
| Mobile Changes | Inserts, Updates Deletes | | | | |
| Filter (Custom) | %SALES_FORECAST_SELECT% | | | | |
| Columns | Server Column | Server Type | Mobile Column | Internal Data Type | Others |
| | Company | String | COMPANY | String | |
| | CompanySynonym | String | COMPANYSYN | String | |
| | Lastname | String | LASTNAME | String | |
| | Firstname | String | FIRSTNAME | String | |
| | PartnerInt | String | PARTNERINT | String | |
| | Contact2 | String | CONTACT2 | String | |
| | From | String | NOTEDBY | String | |
| | ComposedDate | Date/Time | COMPOSEDDATE | String | Res: Day |
| | ForecastDate | Date/Time | DATEFORECAST | String | Res: Day |
| | QuoteDate | Date/Time | DATEQUOTE | String | Res: Day |
| | OrderDate | Date/Time | DATEORDER | String | Res: Day |
| | AmountForecastText | String | AMOUNTFORECAST | String | |
| | AmountQuoteText | String | AMOUNTQUOTE | String | |
| | AmountOrderText | String | AMOUNTQORDER | String | |
| | ResellerName | String | RESELLER | String | |
| | DistributorName | String | DISTRIBUTOR | String | |
| | Forecast_Product | String | PRODUCT | String | |
| | ForecastProductType | String | PRODUCTTYPE | String | |
| | ForecastVLP | String | VLPTYPE | String | |
| | ForecastBand | String | BANDING | String | |

| | | | | | |
|------------------------|------------------------------|--------------|--------------------|------------------|--------------------------|
| Columns (continued) | ConsiderationFactor | String | CONSIDERATI ON | String | |
| | CancelReason | String | REASONLOST | String | |
| | DateLost | String | DATELOST | String | |
| | PDA_Subject | String | SUBJECT | String | |
| | AddressUNID | String | CONTACTID | String | |
| | Comments | RichText | COMMENTS | String | |
| | Form | String | FORM | String | |
| | AccessReader | String | ACCESSREAD ER | String | |
| | AccessAuthor | String | ACCESSAUTH OR | String | |
| | DocID | String | CONTACTID | String | Primary Key |
| | PDA_UpdatedBy | String | UPDATEDBY | String | |
| | PDA_CreateDate | Date/Time | CREATEDATE | Date/Time | Res: Day |
| | PDA_CreateTime | Date/Time | CREATETIME | Date/Time | Res:Second, Time only |
| | WfTask | String | WFTASK | String | |
| | WfReviewDate | Date/Time | WFREVIEWDAT E | Date/Time | Res: Day |
| | WfReviewDateNotifi cation | String | WFREVIEWNO TIFY | String | |
| | PDA_wfState | String | WFSTATE | String | |
| | PDA_wfNewType | String | WFNEWTYP E | String | |
| PDA_wfNewTask | String | WFNEWTASK | String | | |
| | PDA_wfNewMembe r | String | WFNEWMEMB ER | String | |
| | PDA_wfNewTitle | String | WFNEWTITL E | String | |
| | PDA_wfNewNotifica tion | String | WFNEWNOTIF Y | String | |
| | PDA_wfNextXOrTa sks | String | WFXORBRANC HES | String | |
| | PDA_wfNextOrTask s | String | WFORBRANCH ES | String | |
| DB2e Info | Column | Name | Data Type | Length/ Scale | Key |
| | 0 | CREATEDATE | Date | 4/2 | |
| | 1 | CREATETIME | Time | 4/2 | |
| | 2 | COMPANY | VarChar | 50/2 | |
| | 3 | COMPANYSYN | VarChar | 50/2 | |
| | 4 | LASTNAME | VarChar | 30/2 | |
| | 5 | FIRSTNAME | VarChar | 30/2 | |
| | 6 | CONTACTID | VarChar | 33/2 | |
| | 7 | DOCUMENTID | VarChar | 33/2 | Yes |
| | 8 | SUBJECT | VarChar | 100/2 | |
| | 9 | COMMENTS | VarChar | 2000/2 | |
| | 10 | FORM | VarChar | 20/2 | |
| | 11 | ACCESSREADER | VarChar | 250/2 | |
| | 12 | ACCESSAUTHOR | VarChar | 250/2 | |
| | 13 | WFTASK | VarChar | 50/2 | |

| | | | | | |
|--------------------------|----|-------------------|-----------|--------|--|
| DB2e Info (continued) | 14 | WFSTATE | Character | 4/2 | |
| | 15 | WFREVIEWDATE | Date | 4/2 | |
| | 16 | WFREVIEWNOTIFY | Character | 1/2 | |
| | 17 | WFNEWTTYPE | Character | 1/2 | |
| | 18 | WFNEWTASK | VarChar | 1000/2 | |
| | 19 | WFNEWMEMBER | VarChar | 200/2 | |
| | 20 | WFNEWTITLE | VarChar | 50/2 | |
| | 21 | WFNEWNOTIFY | Character | 1/2 | |
| | 22 | WFXORBRANCHES | VarChar | 2000/2 | |
| | 23 | WFORBRANCHES | VarChar | 2000/2 | |
| | 24 | SALESPERSON | VarChar | 50/2 | |
| | 25 | CONTACT2 | VarChar | 50/2 | |
| | 26 | PRODUCT | VarChar | 50/2 | |
| | 27 | PRODUCTTYPE | VarChar | 10/2 | |
| | 28 | CONSIDERATION | VarChar | 4/2 | |
| | 29 | VLPTYPE | VarChar | 10/2 | |
| | 30 | BANDING | VarChar | 5/2 | |
| | 31 | AMOUNTFORECAST | VarChar | 20/2 | |
| | 32 | AMOUNTQUOTE | VarChar | 20/2 | |
| | 33 | AMOUNTORDER | VarChar | 20/2 | |
| | 34 | DATEFORECAST | Date | 4/2 | |
| | 35 | DATEQUOTE | Date | 4/2 | |
| | 36 | DATEORDER | Date | 4/2 | |
| | 37 | RESELLER | VarChar | 50/2 | |
| | 38 | DISTRIBUTOR | VarChar | 50/2 | |
| | 39 | DATELOST | Date | 4/2 | |
| | 40 | CONSIDERATIONLOST | VarChar | 4/2 | |
| | 41 | REASONLOST | VarChar | 100/2 | |
| | 42 | COMPOSEDDATE | Date | 4/2 | |
| | 43 | UPDATEDBY | VarChar | 50/2 | |

Sales Settings

| | |
|-----------------|---|
| Action | Synchronise Server To Mobile |
| Server Type | Lotus Domino/Notes |
| Database Type | Notes Database |
| Server Name | Full hierarchical name of your Domino server |
| Database Name | Path and file name of the PAVONE Sales Settings database |
| Form Name | Setting |
| Server Changes | - |
| Plugin Name | DB2e Plugin |
| Table Name | ES_Setting |
| Mobile Changes | Inserts, Updates Deletes |
| Filter (Custom) | SELECT @Contains(Form;'Setting') & @Contains(Subject;'Country':'Sales Standard Tasks') |

| Columns | Server Column | | Server Type | Mobile Column | Internal Data Type | Others |
|-----------|-------------------|-----------|-------------|---------------|--------------------|-------------|
| | Subject | | String | NAME | String | Primary Key |
| | ValueText | | String | VALUE | String | |
| | ValueSynchronText | | String | VALUESYNC | String | |
| DB2e Info | Column | Name | Data Type | Length/Scale | Key | |
| | 0 | NAME | VarChar | 100/2 | Yes | |
| | 1 | VALUE | VarChar | 10000/2 | | |
| | 2 | VALUESYNC | VarChar | 10000/2 | | |

Sales Favourite List

| Action | Synchronise Server To Mobile | | | | | |
|----------------|---|---------|-------------|---------------|--------------------|-------------|
| Server Type | Lotus Domino/Notes | | | | | |
| Database Type | Notes Database | | | | | |
| Server Name | Full hierarchical name of your Domino server | | | | | |
| Database Name | Path and file name of the PAVONE Sales Application database | | | | | |
| Form Name | Setting | | | | | |
| Server Changes | - | | | | | |
| Plugin Name | DB2e Plugin | | | | | |
| Table Name | ES_Orga | | | | | |
| Mobile Changes | Inserts, Updates Deletes | | | | | |
| Filter | On Server Form = \$PavoneSettingPersonal On Server Username = %\$NOTESAUTH_NAME% | | | | | |
| Columns | Server Column | | Server Type | Mobile Column | Internal Data Type | Others |
| | Subject | | String | NAME | String | Primary Key |
| | PDAFavouriteText | | String | MEMBERS | String | |
| | Form | | String | | | |
| | Username | | String | | | |
| DB2e Info | Column | Name | Data Type | Length/Scale | Key | |
| | 0 | NAME | VarChar | 100/2 | Yes | |
| | 1 | MEMBERS | VarChar | 10000/2 | | |

Sales Startable Processes

| | |
|---------------|--|
| Action | Synchronise Server To Mobile |
| Server Type | Lotus Domino/Notes |
| Database Type | Notes Database |
| Server Name | Full hierarchical name of your Domino server |

| | | | | | |
|----------------|---|-------------|---------------|--------------------|-------------|
| Database Name | Path and file name of the PAVONE Sales Application database | | | | |
| Form Name | Setting | | | | |
| Server Changes | - | | | | |
| Plugin Name | DB2e Plugin | | | | |
| Table Name | ES_Process | | | | |
| Mobile Changes | Inserts, Updates Deletes | | | | |
| Filter | On Server Form = \$PavoneSettingPersonal On Server Username = %\$NOTESAUTH_NAME% | | | | |
| Columns | Server Column | Server Type | Mobile Column | Internal Data Type | Others |
| | Subject | String | NAME | String | Primary Key |
| | PDAProcessDef | String | PROCESSDEF | String | |
| | Form | String | | | |
| | Username | String | | | |
| DB2e Info | Column | Name | Data Type | Length/Scale | Key |
| | 0 | NAME | VarChar | 100/2 | Yes |
| | 1 | PROCESSDEF | VarChar | 10000/2 | |