University of
Innsbruck

Department of Information Systems, Production and Logistics
Research Unit Quality Engineering

Enterprise Application Integration of Personal Information Systems and the ERP System Comarch Semiramis

Master Thesis

(SS2011)

Supervisor: Univ.-Prof. Dr. Ruth Breu

Emanuel Scheiber

February 15, 2012

Abstract

In this thesis we investigate the integration of the ERP system Comarch Semiramis and personal information management. We develop an enterprise application integration solution to synergize the ERP system with different personal informations systems. The methodology applied for this master project, is the design science research approach. The software solution is implemented in form of a first prototype covering the most important functionalities. In order to define the requirements, creating use cases, and prioritizing the importance of the functionalities, an extensive requirement analysis involving stakeholders will be performed. Based on that, the prototype is designed and implemented. The implementation is evaluated by testing, while the project is assessed by a design science research checklist. Finally, the results are summarized and an outlook for future work is presented.

Acknowledgments

I would like to thank my supervisor, Univ.-Prof. Dr. Ruth Breu, for providing herself as supervisor. Further, I would like to thank my co-supervisor, Dr. Michael, for his constant dedication, support and advice during this thesis. I would like to thank the Comarch team for accompanying me during this project and supplying me with technical knowhow. I would also like to thank my family, girlfriend and friends for their patience and constant support.

Contents

List of Abbreviations

API	Application Programming Interface
APS	Advanced Planning and Scheduling
BIS	Business Information Systems
BO	Business Object
BP	Business Process
CRM	Customer Relationship Management
CSV	Comma Separated Values
DBMS	Database Management System
ERP	Enterprise Resource Planning
GUI	Graphical User Interface
IDE	Integrated Development Environment
IETF	Internet Engineering Task Force
IM	Information Management
IS	Information Systems
IT	Information Technology
ODBC	Open Database Connectivity
OLAP	Online Analytical Processing

OLTP	Online-Transaction-Processing
OMA	Open Mobile Allicance
OQL	Object Query Language
PDA	Personal Digital Assistant
PDI	Personal Data Interchange
PIM	Personal Information Management
PIMS	Personal Information Management System / Software
R&D	Research and Development
SaaS	Software as a Service
SAS	Semiramis Application Server
SCM	Supply Chain Management
SOA	Service-Oriented Application
SyncML	Synchronization Markup Language

List of Figures

List of Tables

1. Introduction

In the following section the topics ERP system and personal information management are introduced. Furthermore, the motivation, goal, and approach of this master project are stated.

1.1. ERP Systems and Personal Information Management

Personal information management (PIM) is nowadays indispensable in case of managing our busy lives. PIM is the study as well practice of people's activities addressing the acquiring, organizing, maintaining, retrieving, using and controlling the distribution of personal related information items (Jones and Teevan, 2007). Such items can be documents, email messages, web links and other things needed for every days use to complete certain tasks or to fulfill a certain role either. In the current information age there exist a lot of computer programs and devices supporting the user's PIM needs. Such programs are mainly called PIM software or system (PIMS), PIM manager, PIM tool or personal organizer. Some of them are also at the same time email client applications or mail user agents, respectively. A further special kind of PIM software with collaboration functionalities, supporting cooperative works, is called groupware (Marca and Bock, 1992). Some well known PIM managers are for example: Google Mail/Calender, Microsoft Office Outlook, OneNote, Lotus Notes and Mozilla Thunderbird with Lightning extension.
On the other hand information management (IM) of organizations is supported by business information systems (BIS). A BIS is a set of interrelated components which cooperatively carry out tasks, including input, output, storage, processing and control actions, with the aim to generate value by converting data into information products to support organizational functions like forecasting, planning, control, coordination, decision making and operational workflow (Bocij et al., 2009).
A special type of business information system is an enterprise resource planning (ERP) system. The term ERP refers to a comprehensive software system approach of integrating a wide spanning set of organizational functions and their corresponding application

programs (Vollmann, 2005).That allows now integrated planning across functions and supports decision making. BIS and ERP systems play an essential role for supporting core processes of an organization and leveraging the process's business value.
The ERP system Semiramis, which is the basis for this master project, is a so called ERP II system. An ERP II systems adapts modern information technologies, meets the challenges of E-business, for example web shop integration, and state-of-the-art interfirm cooperation for supply chain management, e.g. by offering web services (Castillo, 2009) (Bond et al., 2000). Supply chain management is the inter-organizational coordination of material and information flows of the entire value chain process with the aim to optimize this process time and costs (Scholz-Reiter and Jakobza, 1999).

Semiramis is web based, is designed in a modular way and has a graphical user interface accessible via a web browser. The ERP is written in the platform independent programming language Java. Semiramis was completely built from scratch, it is totally web orientated, it was designed to be sector independent, easy adaptable and to suit the needs of small and medium enterprises (Gümbel, 2005).
The main objective of this thesis is to connect the ERP II system Semiramis with personal information managers. It is obvious that businesses have a need for interconnecting disparate systems. For example, an integration adds value to a company's system, for example by optimizing processes. The approach of connecting and integrating different applications is called system integration or in relation to business: enterprise application integration or enterprise system integration, respectively (Prencipe et al., 2005) (Ruh et al., 2001).
Such integration is generally achieved by so called interfaces, in this case a software interface. An interface is an entry point were external applications are able to access to specified resources thus enabling communication of data between the participants. The set of all interfaces provided by a software is called application programming interface (API) (Patterson and Hennessy, 2008). In the case of this project a data integration model is applied by the means of synchronizing data across systems (Ruh et al., 2001). In general synchronization is the harmonization of different processes which may appear concurrently (Engesser et al., 1993). For instance, data synchronization is necessary if two or more semantic equivalent information objects exist on independent systems (Schultz, 2003). Especially, when it comes to collaborative supply chain management the exchange of accurate, complete, and consistent data between business partners is essential (Nakatani et al., 2006). Therefore an intelligent data synchronization is very important to keep a proper data quality (Redman, 1996). That means the data needs to be, for example, accurate, correct and duplicate free (Wang and Strong, 1996). Poor

data quality has huge negative impact on a company and also on supply chain management in general, for example it lowers customer satisfaction, causes bad decision making, and increases costs (Stadtler and Kilger, 2005) (Redman, 1998).

This thesis is structured as follows. In the next sections of this thesis the motivation, aim, scientific approach are described. Then, the most relevant theoretical background is described. Following that, in Chapter 3 existing approaches of ERP system and PIMS integration are listed. In Chapter 4, the results of the requirements analysis are resumed. Afterwards, in Chapter 5 the design and basis technologies for the implemented prototype are covered. In Chapter 6, the implementation of the prototype is sketched. Chapter 7 deals with the testing and evaluation of the implemented solution. Finally, in Chapter 8 a conclusion and an outlook to future work is given. The appendix contains the conducted questionnaire and the user manual of the implemented prototype.

1.2. Motivation

The central question is the issue of synchronizing data, for example partner information or appointments, between Semiramis and personal information managers. First we consider the technical facts. Semiramis offers a bunch of interfaces designed to communicate with external applications. For example, web services and CORBA interfaces. Furthermore, Semiramis supports also export and import functionality of data. On the other hand nearly all current personal information management applications have import and export functions. Many of them also have interfaces where external application can access. For example, Microsoft's Outlook has a so called COM-interface. Mozilla's Thunderbird has a quite similar interface named XPCOM. That means technically it should be possible to synchronize data between these systems. However what is missing is a bridge or an agent, respectively, transferring and translating data from one side to the other.

One main problem is that Semiramis does not support the data models of any PIM software and the same vice versa. The data is currently incompatible and there is currently no relationship. The other problem is that offering interfaces is one thing, but implementing and making use of them by another system is another thing. In fact, Semiramis is not able to use the interfaces of any PIM software natively, and vice versa. That means either Semiramis or the specific PIM software has to be extended

to be able to make use of the opposite interfaces. Alternatively, a middleware could be implemented, which is able to coordinate and orchestrate between the interfaces.
To sum it up at the moment it is not possible to transfer information between Semiramis and a PIM application, without mediations and interventions. The next section describes the aim of the planned solution to this problem.

1.3. Goal

The objective of this master project is to build a bridge and thus to implement an enterprise application integration between Semiramis and PIM systems, by the means of semiautomated data synchronization, and import and export functionalities of PIM data.
This master thesis will encompass the whole process of developing a first prototype including scientific background analysis, business relevance investigation, requirements engineering, design, implementation, deployment, testing, and evaluation. While the produced prototype should contain most important functionalities, the requirements analysis is tackling the broad problem described in the motivation. Additional functionalities may be added by future work, based on the output of this project. However, the prototype should be fully operative.

The process of developing this improvement will involve several stakeholders, users and Semiramis employees likewise. The implementation of this feature should be as Semiramis managed and controlled as possible. That means besides offering bidirectional data exchange, Semiramis should be still recognized as the leading system. The necessary user intervention should be as small as possibly. However, there should be still room for customizing options and possibilities to interfere. For instance, by letting the user choose which duplication detection and error correction method is applied. Furthermore, especially when it comes to export data from Semiramis, authorization policies have to be involved. Therefore, a user should only be able to gather data, for which he has permission. Requirements are dealt with in more detail in Chapter 4. In the following section the scientific approach of this master project is described in more detail.

1.4. Approach

The emphasis of the project's direction encompasses mainly the research discipline information systems and also the related science fields business economics and computer science. Research in the discipline information systems (IS) is mainly leaded by two complementary but distinct paradigms which are behavioral sciences and design sciences (March and Smith, 1995). One the one hand behavioral science in IS is about recognize a business need, and develop and justify theories, in order to explain or predict the phenomena related to this need (Hevner and Chatterjee, 2010). Design science on the other hand is about solving a problem by developing an artifact which is created and evaluated. The goal of behavioral science is to find the truth, while design science aims to optimized utility (Hevner and Ram, 2004). In this master's project the design science research approach is applied to integrate Comarch Semiramis and PIM.

Design in IS comprises both the iterative process as set of activities, as well as the resulting products, the artifact (Walls and Widmeyer, 1992). Artifacts in this context are all kinds of human created things related to the design process, like constructs (vocabulary and symbols), models (abstractions and representations), methods (algorithms and practices), instantiations (implemented and prototype systems), or better design theories (Hevner and Chatterjee, 2010). Hevner et al. define design science research as "Design science research is a research paradigm in which a designer answers questions relevant to human problems via the creation of innovative artifacts, thereby contributing new knowledge to the body of scientific evidence. The designed artifacts are both useful and fundamental in understanding that problem. The fundamental principle of design science research is that knowledge and understanding of a design problem and its solution are acquired in the building and application of an artifact." (Hevner and Chatterjee, 2010). In simple words, it is the building of software artifacts in order to solve human problems. However, created artifacts have to be evaluated in order to prove that the problem is solved in an efficient manner, thus providing utility to its users.

In the context of IS design science research Hevner et al. developed seven guidelines, which should help to conduct and evaluate design science research (Hevner and Ram, 2004). These guidelines are shown in Table 1.1.

Table 1.1.: Design science research guidelines in IS, from (Hevner and Ram, 2004).

Guideline	**Description**
Guideline 1: Design as an Artifact	Design science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Guideline 2: Problem relevance	The objective of design science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research rigor	Design science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a search process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of research	Design science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

The master project will comprise the whole structured process of developing the data synchronization extension prototype. Therefore, the methodology of design science is applied in a pragmatic way. Pragmatic because we do not have a problem that has principally not been solved before for other ERP-PIMS combinations. The guidelines will be meet by the following practices:

- Guideline 1: In order to produce a viable artifact, a fully operational first prototype is developed, which will implement not all but a chosen set of functionalities in order to solve the problem.
- Guideline 2: In order to get a broad view of the problem relevance, a requirement analysis involving stakeholders is conducted. Here, also the most important functionalities for the problem at hand, which will be implemented by the first prototype, are determined.

- Guideline 3: The utility, quality and efficacy of the prototype is evaluated through testing by test cases.
- Guideline 4: Research contributions are captured by the designing of the prototype, its coding and generation of all other artifacts written down in this master thesis.
- Guideline 5: The application of rigorous methods in both the construction and evaluation of the design artifact is done by applying well known and established practices and methods.
- Guideline 6: The design as a search process is also met by simplifying the problem into a subproblem which is implemented by the prototype and thus, representing a starting point. The implementation, its code and all other artifacts are provided to the Semiramis developing and consulting team, so that future work builds up on the result.
- Guideline 7: Communication of research is made sure by the presentation of the results and making of the master thesis public.

In order to ensure that the project addresses these guidelines, the design science research checklist is evaluated in Section 7.2.

In the next chapter the theoretical background is presented, which is also import for Guideline 5, since methods and approaches should be based on established and fundamental background.

2. Theoretical Background

In this chapter, the most important technologies, theories, methodologies and process models relevant for the master thesis are presented. First the theory and history of ERP systems in general are described. Next, the ERP system Comarch Semiramis is explained. After that, the disciplines personal information management, enterprise application integration, and data quality are discussed. Finally, the software development process and related activities and models are covered.

2.1. ERP

In the Eleventh Edition of the APICS Dictionary the term ERP is defined as a "framework for organizing, defining, and standardizing the business processes necessary to effectively plan and control an organization so the organization can use its internal knowledge to seek external advantage" (Blackstone et al., 2005). Enterprise resource planning is the managerial approach for utilizing effective and efficient available organizational resources for business processes and thus optimizing the controlling of these.

Before we proceed with the history of ERP, we have to define first the most important terms of the above definitions.

Efficiency is about "doing things right", while effectiveness is about "doing the right things" (Drucker, 1999). The term "resource" can be seen in a broader scope with the meaning of force, source and aid (Gronau, 2004). That includes resources like operating supplies, monetary capital, employees or products. The basic idea of ERP is about seeing business activities and operations of an organization in an integral and holistic point of view, which is supported by rethinking in the direction of business processes (BPs) (Kurbel, 2005). A business process can be defines as an enclosed, repetitive set of interconnected tasks conducted by agents in a temporal or causal sequence in order to fulfill a business purpose, with the objective to create value (Maier et al., 2009). Also important is the term knowledge, especially in the information systems point of

view. Organizational knowledge develops from business events, represented by data, transformed into information and through application and human interaction finally transformed into knowledge (Chaffey and Wood, 2010). This transformation process is the basis of information management, which is dealt in more detail in 2.2. This process can be efficiently supported by information systems like ERP systems. The history of ERP and ERP systems is described in the next subsection.

2.1.1. Development of ERP

The acronym ERP for the term enterprise resource planning was first used and made popular by the information technology research and advisory firm Gartner Group in 1990 with the publication "A Vision of Next Generation MRP II" (Wylie, 1990). The Gartner Group's definition of ERP included a criteria for evaluating the extend of the software's integration within and across various functions of a organization (Jacobs et al., 2007). Therefore, pointing out the importance of cross-functional integration in a business. One of the first modern ERP systems was the R/3 from the German enterprise software corporation SAP AG. It was the first system of its kind implementing a client-server hardware architecture. The three-tier client/server architecture consists of presentation server, application server and database Server. The architecture of such an ERP system is outlined in Subsection 2.1.2. Also worth to mention is an event, which partly responsible of the success of ERP systems. This event is known as the Year 2000 (Y2K) problem, which was pushing the organizations switch to the new ascending ERP systems (Jacobs et al., 2007). Many companies feared that the millennium bug or Y2K bug, errors arising upon changing the year from x99 to x00, would crash their legacy systems.

The emerge of recent Internet technologies and resulting possibilities, in the 21th century, had also a strong impact in the further developing of ERP. Some of these new influences originated from service-oriented application (SOA), software as a serviceb(SaaS), Web 2.0 as well as Semantic Web technologies. SOA is a set of design principles for composing functionalities of business processes, achieved by loose coupled, platform independent and self contained IT services (Bell, 2008). SaaS refers to web-based application software for enterprise and private customers, which is often based on a pay-per-consumption model (Suite, 2001). Web 2.0 is a smorgasbord of different Internet based technologies and applications which enhances the interactive and collaborative functionalities provided to a Internet user and thus leveraging his capabilities (O'Reilly, 2007). The

semantic web can be seen as an extension to the current web by giving information a well defined meaning in order to improve human - computer cooperation (Berners-Lee et al., 2001).

In 2000 the Gartner Group once again came up with a new acronym, this time ERP II (Bond et al., 2000). The term expresses the trend of changing the perspective from vertically integrated organizations, focused on optimizing internal functions, to organizations optimizing the own position in the supply chain and value network, based on core-competencies (Bond et al., 2000). Thus, extending the role of ERP from enterprise optimization to value chain participation and collaborative commerce. Therefore in order to compete prospectively with competitors, organizations have to improving collaborative business interactions within the supply chain and enabling connection of external processes (Ehrenberg and Ewig, 2006). Such functionality, for instance, could be achieved by including outward-facing elements to publish high quality information for the consumption by collaborating partners (Bond et al., 2000). The consequence for an ERP II system architecture is the move from a web-aware, closed and monolithic system to an open, modular and web-based system. Therefore, the previous mentioned technologies play an important role to support ERP II. However despite offering also features for SCM it has to be noted that an ERP II system is neither a SCM system nor an advanced planning and scheduling-system (APS) (Castillo, 2009). A SCM system or APS system is specially designed for SCM functionalities, like SCM spanning production planning or inter-company wide information transparency. They generally support an ERP system.

In the next subsection the general architecture of an ERP system is described in more detail.

2.1.2. ERP Systems

An ERP System is a company wide computer supported BIS utilizing ERP. In many cases when the term ERP is used, it refers often to the system and not to the more general managerial approach. An ERP system is able to model processes of an organization and in many cases automates them through computerization (Kurbel, 2005). It integrates all data and information relevant for an organization and its processes in a centralized database. For human interaction it offers a uniform graphical user interface (GUI). The architecture of an ERP system changes from system to system. Often it is implemented with a multi tier architecture, existing of three layers. These layers are presentation

layer, application layer and database layer (Altekar, 2006). The presentation layer is responsible for the access of clients and the GUI. The application layer contains the business logic. The layer may consist of one or more application systems. An outline of the architecture can be seen in Figure 2.1.

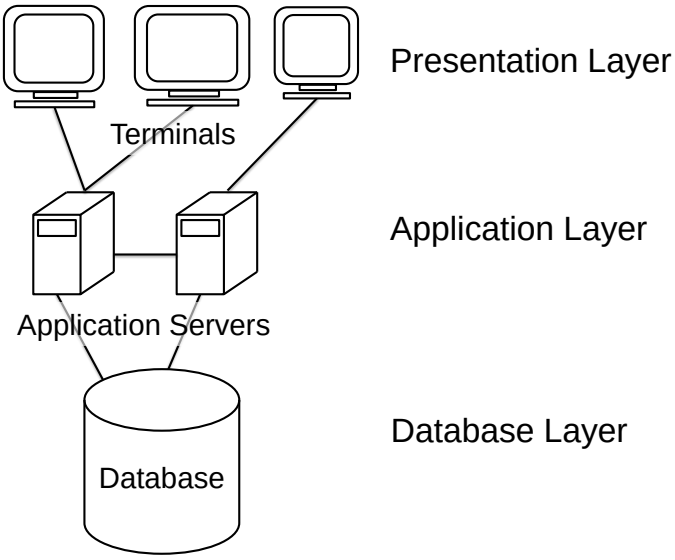


Figure 2.1.: Exemplary 3-tier architecture of an ERP system.

An ERP system stored typically three different kinds of data, which are: customizing data, transaction data, and master data. Customizing data is information of the configuration of the ERP system. For instance, language, time zone, or basis currency. Customizing data is installed at the deployment of the ERP system and is later just seldom changed. Master data is basic business data which is shared for multiple operations and processes. Such data has a long lifetime within an ERP system. Examples of this kind of data are: customer, product, employee, material, or supplier information. On the other hand transaction data is specific data for one instance of a process. Usually this data has a time dimension, refers to master data, and is consecutively numbered. Transaction data is only stored for a specific retention period. Examples are: purchase order, sales return, invoice, or payment records.

The costs of an ERP system generally comprises costs of hardware, software, training, data conversion and reengineering of business processes if needed (Altekar, 2006). The hardware needed to run an ERP system includes personal computers, server computers, network equipment and diverse peripherals. Data conversion costs are the costs which arise when the company has to move information, like customer or supplier records, from

one system to another. With this master project's prototype especially the costs of data conversion can be economized, because of automatization.

In the following subsection, the possible architecture of an ERP II system at the example of Semiramis is described.

2.1.3. Comarch Semiramis

The application integration development of this master thesis focuses on the ERP II system Comarch Semiramis. In this section the system is described in more detail.

The German C.I.S. AG, later under the name Semiramis AG, was founded 1995 in order to develop an ERP system completely from scratch. The system was published in 2003. Semiramis was one of the first ERP II systems for medium sized enterprises. In 2006 C.I.S AG was acquired by the German organization SoftM Software und Beratung AG. SoftM is an IT service provider for small and medium sized enterprises. The main shareholder of SoftM with over 80 percent is the Polish IT corporation Comarch. Comarch is active in the areas telecommunications, finance and banking, the services sector and public administration. Beside distributing Semiramis, the company provides services for IT security, IT architecture, management and outsourcing solutions, customer relationship management (CRM) and sales support, billing, electronic communication and business intelligence.

At the moment the newest version of Semiramis is 5.0. Semiramis offers different modules, also known as frameworks, which are sets of functionalities corresponding to an organizational function or department. These frameworks are: sales, relationship management, purchasing, inventory management, production, planning, calculation, financial accounting, asset accounting, controlling, quality management, agreement management, service, workflow management, business process management, collaboration, document management, statistics, business intelligence, system management and software development. Semiramis was designed to be a cross-industry solution. The current application areas of Semiramis are building materials industry, sheet metal and metal processing, chemical industry, electronics industry, glass and ceramics, wholesale trade, mechanical engineering, plant engineering, tool construction, mechatronics, food and drinks industry and manufacturing industry (Comarch, 2011). Semiramis is multi-language capable. Translations are available for Chinese, Croatian, Czech, Dutch, English, French, German, Italian, Hungarian, Polish, Russian and Slovenian. Furthermore it allows the parallel use of multiple currencies, timezones and units of measurement. Semiramis has a

built-in authorization model, which allows the definition of different user groups and authorization roles. Through authorized partners Comarch offers the Semiramis embracing services: organizational advice, project support, customizing, software implementation, training, installation, hotline service, software maintenance and other IT setup services. Semiramis has an open license model, that means the license can be configured independently of corporate revenue or the number of employees (Gümbel, 2005). Furthermore there are no additional fees for indirect users, which are only using Semiramis indirect, for instance, through third party applications. Semiramis is can also be deployed on-demand or as SaaS solution. The licensing option "Semiramis on demand" allows to adjust license and maintenance costs according to the actual usage.

Semiramis is based on a 3-tier architecture similar to the one described in Section 2.1.2 (Comarch, 2010). It is composed of presentation layer, application layer and database layer. The presentation layer thorough the GUI can be by displayed by an Internet browser. Currently, only Microsoft's browser Internet Explorer is supported to full extent. The presentation layer accesses the application layer through a TCP/IP connection. The application layer, on the mid-level, is realized by so called Semiramis application servers (SAS). This layer represents the business logic. A SAS uses different services of the Semiramis system engine. These are for example: cache-management, transaction service, persistence service, event service, license service, output management, background processing and web server. The communication between the presentation layer and the application layer takes place through the HTTP/HTTPS protocol. The number of SASs within a Semiramis system is dependent on the organization's structure and number of users, because of the restricted resources of a SAS. The database layer is responsible for the data storage. The data is kept in relational databases. Semiramis supports different database management systems (DBMS) like Oracle, MS-SQL-Server or DB2/AS400. The databases are divided in the logically partitions: system configuration database, repository database, online transaction processing (OLTP) database and online analytical processing (OLAP) database. The system configuration database contains central information objects like authorization information, runtime environment configuration. The repository database stores all development objects. These objects define the Semiramis system. Data on this database is only changed when it comes to Semiramis updates and developments. All development objects underlie a strict revision control. Therefore, it is the central point for installation, activation and distribution of software updates and changes. The OLTP database is the storage for (commercial) transaction data and master data. It is the database most stressed in organizational's production and operative processes. One Semiramis system can maintain different OLTP

databases which can be hold isolated from each other. The OLAP database is for the analysis of business data. For this purpose statistics of transaction data and master data are extracted from OLTP databases and stored in OLAP databases. This allows queries over large bulk of integrated data. For the communication with the databases, every DBMS vendor uses own interfaces and SQL dialects, respectively. Therefore to unify communication the object query language (OQL) was implemented. It is based on object orientated and relational concepts, and the database computer language SQL. Furthermore a so called object manager offers database services for developers. Internal the OQL and the object manager are using the platform independent JDBC 3 protocol for communicating with the databases. The overall accesses to the databases is kept as low as possible through an own cache management. For that purpose, relevant data is stored in the computer memory of one's SAS.

Figure 2.2.: System architecture of Comarch Semiramis, from (semiramis.com, 2011).

External application can access the Semiramis system via different interfaces, for example, through WebDAV protocol or web services. There exist web service for data

export, data import, external OQL search and for the possibility to call any Semiramis background applications. The web service protocols currently supported are SOAP and CORBA.

All aspects of a Semiramis system and its architecture are illustrated in Figure 2.2.

2.2. Personal Information Management

Personal information management is a sub discipline of information management. Therefore first the field information management is described in this section. Afterwards, personal information management and personal information manager are explained.

2.2.1. Information Management

Information can be managed in both personal as well in corporate contexts. However the study of its use in the corporate context has led to disciplinary constructs (Middleton, 2006). Information management is very important for businesses. On the one hand the proper management of information can lead to strategic advances for companies, while on the other hand the mismanagement can lead to negative impact on the organization's performance (Porter and Millar, 1985) (Peters and Waterman, 2004).

IM, sometimes also called information resource management, contributes to many disciplines including information science, management science, computing science and information systems. Information management includes the organizing, retrieving, collection, acquiring, distribution and maintaining of information. According to Davenport the process of IM includes: identification of information needs and requirements, acquisition and collection, categorization and storage, packaging and formatting, dissemination and distribution, the analysis and use of information (Davenport, 1993). Another approach of describing the IM process is the life cycle model of information management of Krcmar (Krcmar, 2003). It consists of five management tasks, which are management of: information sources, the information resources, information supply, information demand and the IT infrastructure.

The term "IM" is closely related to data management and knowledge management. In fact the terms data, information and knowledge are ambiguous and often vaguely defined (Lehner and Maier, 1998). However data itself can be seen as carriers, but not the informing content itself (Wijnhoven, 2009). Data which is informative, and

thus information, is meaningful, true and relevant in context of its use (Floridi, 2000). On the other hand knowledge can be seen as contextual information and expert insight. The focus of knowledge management lies on knowledge objects and knowledge processes, which has proven to be difficult to manage (Maier, 2007). Knowledge it is often hard to collect and store, because it is often tacit and embodied (Madhavan and Grover, 1998).

IM in organizations has a broad field of applications. According to Mertens and Knolmayer IM is responsible for: the strategic planning of IT development, leveraging IT for supporting the strategic position of an organization, providing of information as corporate resource,supporting of effective and economical information, managing of technical and human resources (Mertens and Knolmayer, 1998).

A special approach to IM is personal information management, which is depicted in the next subsection.

2.2.2. Personal Information Management

As mentioned before in the introduction, personal information management is the study and practice of people's activities addressing the acquiring, organizing, maintaining, retrieving, using and controlling the distribution of personal related information items (Jones and Teevan, 2007). From the previous definition (compare Subsection 2.2.1) we know that information is data with context. An information item is information which is either written down, recorded or stored else when further used. Examples of personal information items are digital or paper based documents, E-mail records, web links, address entries, captured appointments, and other things needed to complete certain tasks or to fulfill a certain role. While the term personal information may imply other, important is that such items do not necessarily have to be private (Lansdale, 1988). Tasks are not necessarily work related and can also refer to leisure time activities. The roles a human being can fulfill, are for example, employee, parent, member of community, etc.

The phrase "personal information management" came up in the 1980's, in relation to the discussion on how technological solution can simplify our daily lives (Lansdale, 1988). Another important work in showing how to develop theories for improving personal information management practices and tools was the paper "Finding and Reminding" by Barreau and Nardi (Barreau and Nardi, 1995). They studied how users organize and find files on their computers. PIM encompasses many disciplines like cognitive psychology,

human-computer interaction, information retrieval, and library and information science (Jones and Teevan, 2007).

Closely related to the field PIM is personal knowledge management, with overlapping research contributions. Basically personal knowledge management adds the theories created within knowledge management (Grundspenkis, 2007).

The next subsection describes personal information manager, the instruments which help us to manage our personal information.

2.2.3. Personal Information Manager

There exist numerous tools to support PIM, for example: paper documents, books, index card catalog, calendars, Rolodex, and so on. The first comprehensive tool, optimized for real PIM, was the personal organizer. With the advent of personal computers in the 1980's, which made computers attractive for individuals, followed the arise of software supporting PIM needs. Such electronic personal organizer are mainly called personal information manager, PIM software or PIM systems (Bocij et al., 2009). A further important innovation in relation to PIM was the personal digital assistant (PDA), which was first introduced by the computer manufacturer Apple in 1992. A PDA is a mobile device providing functionalities of a personal information manager. Today all modern mobile phones and computers have some PIM capabilities natively integrated.

Basically, a personal information manager supports the users at storing, organizing and retrieving personal information. It allows the user to schedule appointments, share information, setup reminders, record information, track expenses, organize files, create task lists, and much more. Some sophisticated managers can have also included following functionalities: email client, project management, synchronization with other tools and devices, document management, news subscription, customer relationship management, knowledge database or chatting functions. PIM software with collaboration functionalities that is supporting cooperative works, is called groupware (Marca and Bock, 1992). In Subsection 4.1.2 the market of PIM mangers is studied in more detail.

When information gets scattered across different devices and systems we speak of "information fragmentation" (O'Brien and Al-Soufi, 1993). Especially, PIM is very vulnerable to information fragmentation. The reason is that many people use different computers, at home and work, use mobile phones, PDA or analog devices and web applications.

Furthermore, it is also likely that they will change the tools after some time. Applications have often their own data format for storing information items which leads to a heterogeneity of data. The result is that user's personal information is scattered across computers and applications, therefore we can speak of personal information fragmentation (Jones, 2007). Users have to consider different resources to get whole picture of their relevant information. On large scale, for instance, in big organizations, this is even much more a problem, because users have to know that such resources even exist (Wood, 2010).

The solution to information fragmentation and heterogeneity of data is integration of data and applications, which is the main topic of the next section.

2.3. Enterprise Application Integration

In the previous section it was described how information fragmentation and heterogeneity of data can evolve. These reasons are the motivation for implementing an integration. In fact, there are many approaches to integration: spanning from enterprise, business, application and system integration. Enterprise integration is the most comprehensive term, comprising all other approaches (Bernus et al., 1996). While enterprise integration is on an organizational level, business integration level is on the process level. However, in this master thesis we concentrate on system and application integration.

Basically, system integration is about connecting different systems into one symbiotic unifying system. The business of system integration goes beyond the engineering level (Prencipe et al., 2005). On the one hand it refers to the internal activities of organization of integrating inputs and outputs, of business processes, to produce new products or other business value. On the other hand it also refers to external activities of integrating components, skills, and knowledge of other partners, including stakeholders, suppliers, employees and customers. Therefore it is quite important in supply chain management, customer relationship management, business intelligence, human resources, and other business aspects.

There are several integration techniques in order to achieve system integration, like business process management, programming, networking, and especially enterprise application integration (EAI). EAI describes a comprehensive approach for integrating application systems. It allows the exchange of information between applications in organizations and also beyond a firm's border, without greater changes of the existent

systems (Kaib, 2002). In practice that means EAI represents a set of frameworks, principles and technologies for achieving system integration. EAI's approaches are mainly implemented through a middleware. Middlewere can be either refer to technology providing application-independent services that mediate between applications, or software products implementing these services (Ruh et al., 2001).

There are three points in an application where an integration can take place: in presentation, data and functional layer (Ruh et al., 2001). Resulting in three integration models: presentation, data and functional integration models. Each integration model defines the characteristics and mechanisms for the integration of applications on a certain level. In the case of presentation integration model, multiple existing program components are merged by offering one unifying graphical user interface. For instance, the user can navigate through multiple applications with a single GUI. A functional integration model allows software to invoke external functionality and business logic, which is implemented in other applications. Thus it allows a much deeper integration than on presentation level. Such an enactment of external functions is often supported by APIs.

In this master thesis, however, we concentrate on the data-level of enterprise application integration, which is described in more detail in the following subsection.

2.3.1. Data Integration

The basic idea of the data integration model is to bypass the presentation and business logic and thus directly extract or load data in databases and data structures of an application (Linthicum, 2000) (Ruh et al., 2001). The role of EAI on the data level can be seen in Figure 2.3.

The terminology around data integration is wide spanning, in this paragraph the most important terms are introduced. The term data integration itself is not only used in the context of combining data located at different sources, also in the context of providing users a unified view of these data (Lenzerini, 2002). For example, the process called enterprise information integration deals especially with the latter context. It is a process using data abstraction for providing uniform data access, in order to let heterogeneous data sources appear, to users and systems, as a single homogeneous data source (Halevy et al., 2005). When it comes to information or knowledge (compare Subsection 2.2.1), the terms information integration or knowledge integration, respectively, are often used instead of data integration (Hearst, 1998) (Jetter and Kraaijenbrink, 2006). In context

of the semantic web, the term semantic integration is relevant. It is about semantically interconnecting information of different sources (Guido and Paiano, 2010).

There are many scenarios where it makes sense to use the data integration model. In general it is often used for the purpose of reusing or synchronizing data across applications. Especially for analysis, decision making and for other statistical purposes it is often needed to combine data from multiple sources. Furthermore, when data is reformatted and updated at one source, data integration allows to make sure that the data is synchronized and consistent at all databases, where the data is kept.

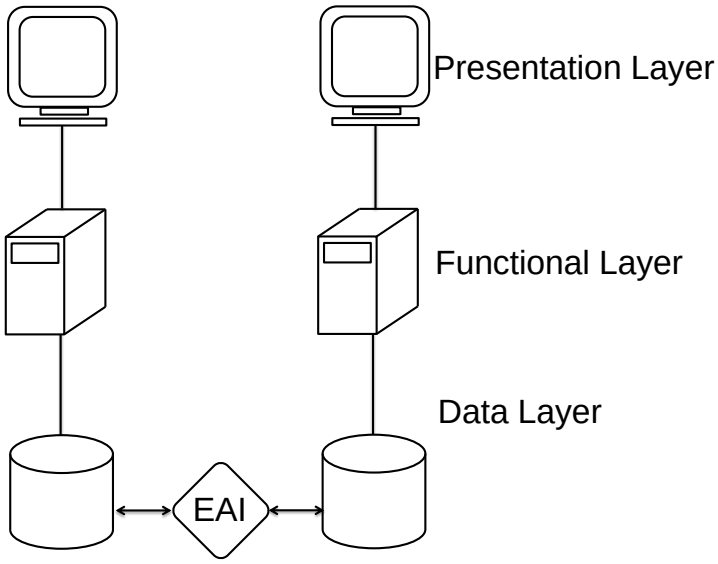


Figure 2.3.: The role of EAI in the data integration model.

There exist a lot of techniques to work with data integration. Often data warehouses are maintained in order to provide data to multiple applications. Such data warehouses are big databases containing data of multiple databases, which are integrated through data integration. Data mining and OLAP software are used to access data warehouses and to process the data. Both OLAP and data mining technologies are important for analytical researches in businesses. While OLAP is about the resolution of multi-dimensional analytical queries, the priority of data mining is the extraction of patterns from large data sets (Nisbet et al., 2009). Other technologies useful for data integration are batch file transfer, open database connectivity (ODBC), data transformation and database access middleware. Batch file transfer is for moving data between databases. ODBC is an API for the abstract access to heterogeneous relational databases. Data transformation is important to convert data from one format into another one. For instance, the language

XLST is used for transformation of XML documents. XML is a wide spread standard for representing data and information. Database access middleware are tools which provide connectivity to distributed databases, enable the common management of results and allow the exchange of queries.

Data integration allows us to have information at the right place, however it is very important to have the right information and at right time. Thus information and data has to fulfill certain quality standards. In the next section the criteria and measurements to achieve such are explained.

2.4. Data Quality

Data integration enables the supply and access of information objects needed for certain functions. However these information objects have to meet functional and agents specific requirements and quality standards (Jung, 2006). Quality in this context can be defined as "fitness for use", i.e., data fit for utilization by data consumers (Tayi and Ballou, 1998) (Wang and Strong, 1996). This also implies that the concept is relative and subjective (Tayi and Ballou, 1998). The term information quality refers to the information definition (compare Subsection 2.2.1) and is often used as synonym for data quality in context of information systems (Wang et al., 2005). For rating the quality of data, data quality can be broken down to as set of quality attributes which represent single aspects or constructs of data quality (Wang and Strong, 1996).These attributes are called data quality dimensions.

In the following enumeration all 15 dimensions are explained after Wang et al. (Wang and Strong, 1996) (Wang et al., 2005) (Pipino et al., 2002):

- Accuracy or free-of-error describes the extent to which data is correct.
- Believability expresses the extent to which data can be considered as reliable and true.
- Objectivity is the dimension to which data is impartial, unprejudiced, and unbiased.
- Reputation is the extent to which data is eminently respectable in context of its content or source.
- Relevancy describes the extent to which data is usable and beneficial for the task at hand.
- Timeliness or currency is the extend to which the data is sufficiently up-to-date.

- Value-added expresses the extent to which data is profitable and provides advantages for its use.
- Completeness highlights the extent to which data is not lacking and its breadth and depth are sufficient for the task at hand.
- Appropriate amount of data is the dimension to which the volume of data is adequate for the task at hand.
- Interpretability describes the extent to which data is clear defined and in adequate representation (language, symbols and units).
- Ease of understanding or understandability is the extent to which data can be easily understood.
- The representational consistency describes the extent to which data is represented in the same format.
- Concise representation is the extend to which data is represented compactly.
- Ease of manipulation is the extent to which data is easy to manipulate and can be applied to different tasks.
- Accessibility describes the extend to which data is available, thus how easily and quickly retrievable it is.
- Security of data is the extend to which the data's access is restricted adequate to assure its security.

In the next section we will focus on the measures to improve and maintain data quality, in relation to data integration.

2.4.1. Data Quality Measures

There are many data and information quality measures and approaches for managing data quality in order to improve data quality, which can be applied in each stage of the information management life cycle (compare Subsection 2.2.1). In the following paragraph the most important concepts are explained.

A method to clearly define data objects, and thus creating a standard to control and check against, is the so called data dictionary. A data dictionary contains structure definitions of data, including their format, relations, attributes and allowed values (Chaffey and Wood, 2010).

Data validation is the process, running while data is entered into a database, which ensures only valid and correct data is stored. It also monitors the security of the data. Data validation rules can be implemented with the help of a data dictionary.

Before data quality can be improved once it is in the database, first the level of data quality has to be determined, this process is called data quality auditing. Data quality auditing is a structured approach using database queries and also manual examining to assess data quality (Chaffey and Wood, 2010). A similar method is data profiling, which is a data analysis to rate the data quality of the data's attributes (Chaffey and Wood, 2010).

Data quality can be assessed with the help of metrics. There exist several strategies to develop metrics. A metric could be based on logical considerations, for example, rule validation based on logical deduction (Wu and Su, 1993). For instance a possible result could claim that a data attribute is either consistent or not consistent. On the other hand a metric could be also rated founded on probability theory. For instance, a metric for timeliness or currency, respectively, could express the probability of a data attribute value corresponding to its real world counterpart (Heinrich et al., 2009).

The automatized process of improving the data quality itself is called data cleansing, which is often applied to customer records in an organization (Chaffey and Wood, 2010). The two main forms of data cleansing are de-duplication and hygiene processing. Hygiene processing deals with the data format and the attributes allowed values. Usually hygiene processing consists of the three steps: formatting, parsing, and validation with enhancement (Chaffey and Wood, 2010). Formatting removes, for example, unnecessary blank space. Parsing identifies main data elements, which can be located in combined fields. For instance the ZIP code and city are often stored in one field. Validation for instance checks if the value of an attribute is in the right format or has the right values. If this is not the case, errors may be enhanced automatically if possible. On the other hand de-duplication is the process of matching data fields and identifying potential duplicates and removing them. The optimal goal of de-duplication is to minimize both overkill and underkill (Chaffey and Wood, 2010). Thus avoiding the deletion of data not being identically and avoid the retain of duplicates.

However in the scope of this master project we mainly concentrate on the measures needed when it comes to data integration, which is for the most parts duplicate detection, which is described in more detail in the next subsection

2.4.2. Duplicate Detection

Duplicate detection or record linkage refers to the process of detecting and combining information from two records that are believed to relate to the same entity, thus are

semantically equivalent, and are a match or link, respectively (Herzog et al., 2007). A match can be either an exact match or statistical match. An exact match indicates that both entities are really the same one. A statistical match states that two entities are to a certain degree equal, so that they can be combined. However in this master project only exact matches are regarded, since we want an error free approach of record linkage, as far it is possible.

The idea of record linkage dates back to Halbert L. Dunn in 1946 (Dunn, 1946). He also relates the process to PIM integration. In his paper Dunn speaks about creating a "Book of Life", which is made up of the records of all the major events in a person's life. Dunn claims that the role of record linkage is "assembling the pages of this Book, into a volume"(Dunn, 1946).

Record linkage is known under many other names like: merge/purge processing, list washing, data/record matching, object identity problem, entity resolution, identity resolution, entity disambiguation, duplicate detection, record matching, instance identification, coreference resolution, reference reconciliation, and database hardening (Elmagarmid et al., 2007). Based on record linkage is the process data fusion or information fusion. Data fusion is the merging, the completion, and combination of incomplete data objects from different sources (Torra, 2003). Thus, data fusion reduces the size of the total data set.

Usually the process of record linkage comprises the steps: preprocessing of the data and determination of parameters, formatting and partition of the data, detection of duplicates, and fusion and integration of data (Apel et al., 2009). The determination of parameters may include, depending on the strategy, the generation of the match keys, assignment of thresholds, agreement patterns etc.

The process of record linkage is very challenging if records have no unique fields for identification, for example, social security number of a person. It is even harder if the data and its attributes are stored in different formats. However there are many strategies and algorithms for supporting record linkage, the most important ones for exact matching are based on deterministic and probabilistic methods (Herzog et al., 2007).

In deterministic record linkage, two data entries are declared to be a link, thus are the same, if a collection of identifiers are exactly identically (Herzog et al., 2007). Such a set of identifiers is called the match key. For instance, for matching addresses the match key could be a combination of ZIP code, street number and first few letters of the street name.

A popular application of probabilistic record linkage is the Fellegi-Sunter model. It classifies two data objects as either links, possible links or non-links (Fellegi and Sunter, 1969). In this procedure first the two data sets are converted into a standardized format. Then the fields of the records are compared against. With the help of two conditional probabilities a numerical ratio, R, can be defined, which is expressing the ratio of a pair of records, r. The formula is:

$$R = \frac{P(\gamma \in \Gamma \mid r \in M)}{P(\gamma \in \Gamma \mid r \in U)} \tag{2.1}$$

M is the total set of matching records, while U is the set of non matching ones. γ is an arbitrary agreement pattern in a comparison space Γ. γ can be an agreement pattern states as a set of binaries indicating which field is defined to be equal and which not. Therefore $P(\gamma \in \Gamma \mid r \in M)$ expresses the probability that a record pair r has an agreement pattern γ, given that it is a match. Finally through a decision rule, the ratio can be assigned to classify a pair as match, potential match or non-match. Fellegi and Sunter proposed the decision rule (Fellegi and Sunter, 1969):

- If $R \geq$ Upper , then the pair r is a match.
- If Lower $\leq R \leq$ Upper, then the pair r is a potential match.
- If $R \leq$ Lower, then the pair r is not a match.

A priori chosen error bounds on false matches and false non-matches, for example, with help of empirical values, are usually used to determine the cutoff thresholds Upper and Lower.

Also proposing a challenge for record linkage are typographical errors and variations. There are several algorithms which can be used to detect such errors. One algorithm dealing with this problem is the Hamming distance, which is a measure of dissimilarity of strings (Hamming, 1950). It simply counts the number of different characters. For example, 322344 and 312342 would have a Hamming distance of two. Very similar is the Levenshtein distance or also called edit distance (Levenshtein, 1966). The Levenshtein distance is also metric, which measures the difference between two strings. It computes the minimum number of edits needed to convert one string into the other. The allowed edit operations are insertion, deletion, or substitution of single characters. Typewriter distance or keyboard distance is also a similarity measure for two strings (Min and Wilson, 1995). Here the distance between two different characters of two comparing

strings is calculated through the distance of the keys on the computer keyboard. For instance, the key 'q' and 'w' have a distance of one, while the distance between the keys 'q' and 'e' is two. Further approaches are phonetic algorithms, which are indexing words by their pronunciation (Kessler, 2005). An example of a phonetic algorithm is Soundex, which is especially design for the English language (The National Archives, 2007). On the other hand the Kölner Phonetik is a phonetic algorithm suitable for the German language (Postel, 2005). Even the Internet can be used support the matching of two entities, such an approach is called web based linkage (Elmacioglu et al., 2007). Sometimes it makes also sense to include synonyms of words when comparing string. For instance the project WordNet provides therefore a open word database (Fellbaum, 1998).

There are several ways to implement a duplicate detection algorithm. The naive approach would be to implement it with two nested loops running over all data entries. A better method is to group and partition the datasets into blocks and only examine duplicates within them. The data attribute to group should be already as error free as possible. For example, group by the first three numbers of the ZIP field. Another similar approach of improving the efficiency of record linkage is the so called sorted neighborhood (Yan et al., 2007) (Monge and Elkan, 1997). Here the basic idea is to first sort the records so that potential duplicates are located near each other, thus only the neighborhood of a record has to be examined. To this end a match key is generated and with the match key the entries are sorted. Then a chosen window size is used to determine the neighbors. Both algorithms, blocking and sorted neighborhood, can be performed multiple times with different match keys and groups, respectively, to minimize the false positive and false negative detections.

Evaluation of the results can take place through analyzing the true positive, false positive, true negative and false negative results (Olson and Delen, 2008). The precision, the amount of true duplicates in the results, can be calculated as (Olson and Delen, 2008):

$$\text{precision} = \frac{|\text{truepositives}|}{|\text{truepositives}| + |\text{falsepositives}|} \tag{2.2}$$

Recall, which is the amount of found duplicates among all duplicates, it can be computed as (Olson and Delen, 2008):

$$\text{recall} = \frac{|\text{truepositives}|}{|\text{truepositives}| + |\text{falsenegatives}|} \tag{2.3}$$

The F-measure or F1 score, respectively, the harmonic mean of precision and recall, can be calculated as (Olson and Delen, 2008):

$$\mathrm{F-Measure} = 2 \times \frac{\mathrm{recall} \times \mathrm{precision}}{\mathrm{recall} + \mathrm{precision}} \tag{2.4}$$

The F-measure is a measure for the accuracy of the test and thus is the optimization goal in this case.

2.4.3. Conflict Resolution

Record linkage only helps us to detect duplicates and thus conflicts, for example, we have found matching records where one attribute is different. Now, it has to decided what to do. One solution would be to let the user decide manually. Especially, when it comes to large datasets this could be unpractical. Therefore a conflict resolution policy can determine which data is selected as valid one, and thus conserved. The typical conflict resolution policies are: originator wins, recipient wins, client wins, server wins, recent data wins, duplicate, and ignore (Lee et al., 2004). Originator wins means that the data record of the originator is hold and the data of the other synchronization partner(s) is overwritten, when allowed. The corresponding behavior is also the case at recipient wins, client wins, and server wins. Recent data wins implies that the data most recently updated in time is the valid one. In the case of duplicate, conflicting records are conserved and thus duplicate entries are written in the involved databases. On the other hand, ignore results that the records are not exchanged at all if a conflict arises.

For this master project not only concept and methods of the enterprise application integration and data quality are important but also the development itself. Since a solution is implemented, it is also necessary to describe the software development process, so that one can follow and understand the proceeding of this thesis.

2.5. Software Development Process

The software development process methodology used in this master project is the so called waterfall model. A software development process or software development life cycle is a procedure model for representing and structuring the development of software applications. An important methodology for describing a linear software development process is the waterfall model. It is a sequential design process containing several phases,

however their number and notation vary in literature. According to Royce these phases are: requirements for system and software, analysis, program design, coding, testing and operations (Royce, 1987). Iterative interaction, limited to successive steps, between two phases are also possible. In Figure 2.4 the waterfall model by Royce is outlined. In the first stage the system requirements are defined. They point out which hardware and computing power is necessary, thus giving a frame in which the software can operate. The software requirements mainly define which functionality should be implemented. In the analysis phase it is evaluated if the requirement can be met and what will be needed to implement the software. In the program design step the architecture and other implementation technical aspects, like algorithms, of the software are outlined and modeled. The design proposes a possible solution to the implementation of the software. The coding phase deals with the real implementation and programming of the application. In the testing stage the quality of the software is determined, by running tests and evaluating them. Finally the operations stage comprises the delivery, deployment, updating, monitoring and maintenance of the software. For the most part, errors are fixed in this phase.

The waterfall model has received a lot of criticism (Boehm, 1988). Mainly for not being dynamic, agile and responsive, to requirement modifications or changing customer needs and expectations. However since this project is a time limited, one-man and self-contained project it makes sense to use this rigid model.

In the next subsection the phases are described in more detail.

2.5.1. Phases of the Waterfall Model

The first three phases of the waterfall model can be covered by a requirements analysis or requirement engineering, respectively (Leonard and Press, 1999) (Laplante, 2009). The requirements analysis determines the needs of the application to develop, taking into account the requirements of the underlying computer system and various stakeholders. Requirements are categorized into functional and non-functional requirements. Functional requirements express what the software has to do, while the non-functional requirements state how well and to which degree the functions should be covered. According to Leonard good requirements have to fulfill certain criteria (Leonard and Press, 1999). A requirement has to be achievable at reasonable costs. Furthermore, a requirement has to be verifiable, for instance, so that the expected performance is quantitative. It must be unambiguous so that it has no other possible meaning. Requirements have to

be complete and should contain all information necessary to apprehend the customer's need.

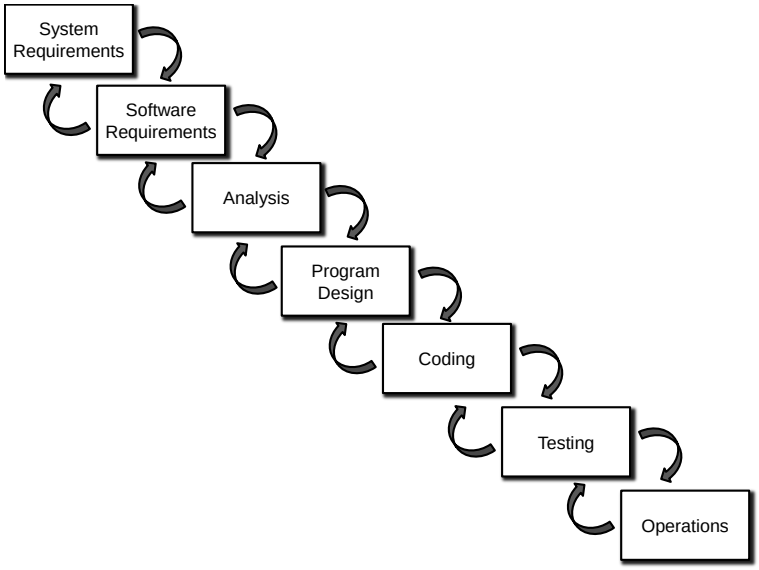


Figure 2.4.: Waterfall model by Royce (Royce, 1987).

A requirement should be defined in terms of need and not solution. Thus, it should be about the why and what, and not the how. Additionally, they should be consistent with other ones Conflicts have to be avoided from the beginning. The output of the requirement analysis should be either expressed in operational, functional or physical view (Leonard and Press, 1999). The operational view is related to the user's expectation, describing how it will serve. For instance, it says how well or under what condition the application operates. On the other hand the functional view concentrates on what the application has to do in order to produce the desired operational behavior. For example, it addresses required inputs, outputs, states and transformation rules. The physical view deals with how the system and software is constructed. It focuses on the interfaces between operators and equipment, and technology requirements.

The program design phase comprises the planning and modeling of the software solution. Here it is defined how the software is implemented, including measures regarding reusability, security, usability, maintainability.

The coding phase deals with the implementation of the software design. Generally it includes the writing of the source code, pre-testing, and debugging. Debugging is about finding errors and resolving them.

In the testing phase the evaluation and verification of the software takes places. Testing is an activity evaluating and improving the quality of the developed product, mainly by capturing problems and defects (IEEE Computer Society, 2004). For testing the functionality of a software, test cases are executed and compared against expected behavior. A test case is a sequence of activities and set of conditions with them a tester can test a software if it is working properly (Alexander and Maiden, 2004). In the case of scenario testing, such test cases for the functionality can be derived from use cases, usage scenarios or user stories. User stories are mainly originating from the customer when defining the functional requirements of the software (Cohn, 2004). It is an informal or formal functionality description expressed with one or two sentences including a role and priority. For example: "As a user, I want to synchronize my address data so that no duplicates are generated". On the other hand a usage scenario describes one concrete example by describing all activities of the example sequence during the system-user interaction (Alexander and Maiden, 2004). They are very similar to the test cases. One usage scenario covers generally one test case. An use case in contrast is describing a process and all its steps between users and a software in detail (Alexander and Maiden, 2004). Thus its encompass all possible scenarios of the process. It also includes how the goals are achieved and how exceptions are handled. It is a software modeling technique and much more extensive than a user story. A use case may be delivered as a document on its own. One use case may cover multiple test cases. A use case can be illustrated by a use case diagram in the modeling language UML (Booch et al., 2005). A use case diagram displays a set of use cases and actors, and their relationships (Booch et al., 2005). Actors can be either a person, organization, or external system which have interactions with the application. On of the most basic testing method is to walk through beforehand defined usages scenarios and use cases. Such walkthroughs are belong to so called doneness tests. Doneness test can be used to assure that the requirements, like use cases, are met to a predefined degree of precision and correctness (Alexander and Maiden, 2004).

The operation phase deals with the deployment and maintenance of the software. The deployment is the installation and configuration of the software, and thus making it available to the end users. Lientz and Swanson categorized maintenance activities into four classes (Lientz and Swanson, 1980): adaptive, perfective, corrective, and preventive activities. Adaptive maintenance activities cope with changes and adjusting in the software environment. When adapting to additional or changed user requirements we speak of perfective activities. Corrective activities are about handling and fixing found errors. On the other hand preventive activities deals with the prevention of potential problems

and improving the maintainability of the software.

3. Existing EAI Approaches of PIM and ERP

In this chapter we review selected software applications related to the problem at hand. Therefore software bridging the gab between PIMS and ERP systems are examined.

The extensive Internet research has shown that several ERP systems allow the integration of Microsoft Outlook, either natively or with add-ons. However seldom for any other PIMS, when the vendors do not have a own PIMS in their portfolio. For example, which is the case at IBM with its ERP OpenPro and its PIM solution Lotus Notes. The reason why Microsoft Outlook is supported so well is, that is has by far the highest market share. More details on the market share of PIMSs are given in Subsection 4.1.2. Nearly all integration solutions have the core functionality of synchronizing contact data, calendar data, and the possibility for the attachment of Emails.

To give a broad overview, the Table 3.1 lists several promising solutions found during the Internet research. The considered criteria for the table are functionalities, conflict resolution and implementation. These are the most important facts to learn how other software have already solved the same problem as dealt within this master thesis. The functionalities and conflict resolution options tell much about what is possible at all. The implementation give useful ideas on how these functionalities can be achieved. However, further interesting details, like record linkage methods or interface implementation, are lacking, because these are seldom revealed by the developers.

Table 3.1.: ERP and PIMS integration solutions.

Solution	**SAP Business One Add-on Microsoft Outlook Integration**
ERP system	SAP Business One
Supported PIMS	Microsoft Outlook
Functionalities	Synchronization of contacts, tasks and calendar; attachment of Outlook Emails, Microsoft Word files and Microsoft Excel; generation of quotations in Outlook for later import in SAP.
Conflict Resolution	Synchronization preview for manual resolution; selection of dominant system and direction
Implementation	Client side, Outlook add in

Solution	**Microsoft Dynamics AX Outlook integration**
ERP system	Microsoft Dynamics AX
Supported PIMS	Microsoft Outlook
Functionalities	Synchronization of contacts, appointments, tasks, and activities
Conflict Resolution	Not available
Implementation	ERP system side; Outlook folder access

Solution	**Sage CRM Outlook Integration**
ERP system	Sage CRM
Supported PIMS	Microsoft Outlook
Functionalities	Synchronization of contacts, appointments, tasks, and activities; attachment of Emails; embedding the ERP system in Outlook
Conflict Resolution	Selection of dominant system and direction; can only be set by an administrator and is system wide.
Implementation	Client side; Outlook add in

Solution	**Outlook Add-In for Microsoft Dynamics NAV**
ERP system	Microsoft Dynamics NAV
Supported PIMS	Microsoft Outlook
Functionalities	Synchronization of contacts, appointments, tasks, and activities
Conflict Resolution	Selection of dominant system; conflict details dialog box for manual resolution
Implementation	Client side; Outlook add in

Table 3.1.: ERP and PIMS integration solutions.

Solution	**Lotus Notes Access for SAP Solutions**
ERP system	SAP ERP systems
Supported PIMS	IBM Lotus Notes
Functionalities	Integration into calendars and scheduling, contact management, workflow processing, etc. by access of SAP information and business processes
Conflict Resolution	Only one direction while source system is the dominant system
Implementation	Communication through connection with SAP

Solution	**eNVenta ERP: Office Integration**
ERP system	eNVenta ERP
Supported PIMS	Microsoft Outlook
Functionalities	Synchronization of contacts, calendar and tasks; (automatic) assignment of Email addresses to customer data and tasks; storage of Emails, and other Microsoft Office and Open Office documents
Conflict Resolution	No details known
Implementation	No details known

Solution	**Mozilla Thunderbird and Microsoft Outlook interfaces for OpenERP**
ERP system	OpenERP
Supported PIMS	Mozilla Thunderbird; Microsoft Outlook
Functionalities	Synchronization of contacts and calendar; create a contact or partner from an e-mail; storage of Emails; send any attachment to an OpenERP document
Conflict Resolution	No details known
Implementation	Client side; Outlook add in and Thunderbird add in

In the following some popular solutions are represented in more detail. The reason why only Outlook integrations are considered is only because these were the best documented applications.

3.1. SAP Business One Add-on Microsoft Outlook Integration

SAP AG one of the largest companies developing ERP systems. SAP's ERP system Business One is a solution for small and medium sized enterprises. Therefore it a competitive product to Semiramis. SAP Business One Add-on Microsoft Outlook Integration allows the exchange of data between SAP Business One and Microsoft Outlook. The core functionality is the data synchronization of contacts, task and calendar. Additionally, it is possible to attach Outlook Emails, Microsoft Word files and Microsoft Excel files as task into Business One. A further added function is the possibility to generate quotations in Outlook without connection to Business One. For instance, if it is done outside of the office. And, later it is possible to import the quotations into Business One.

The user interaction is realized through an Outlook add-in. In fact it is a client side application, which can be installed with a stand alone installer. The Outlook add-in adds graphical buttons in the tool bar for the added features. In order to work with them, first a connection to the ERP system as to be established, which is implemented by a Outlook button in order to log in. These new buttons, for example, allow quick synchronization and quick save of Emails as activity. When editing a contact in Outlook the field "associated contact" is available. This tells the user with which counterpart in the ERP system it is or will be synchronized. Here it is also possible to synchronize the single contact. Here also a customer snapshot can be generated. Such a customer snapshot attaches a spreadsheet to the Outlook contact which holds several informations on the customer, like orders, service calls, account balance and more.

The integration solution has several settings for synchronization. General settings are available for defining which system has to be trusted, thus has priority. It can be set which data should be exchanged and in which direction. It can be chosen if a synchronization preview window is displayed or not. It also can be decided that the conflicts are automatically resolved. When synchronizing the calendar, the time frame of the appointments to be synchronized can be chosen. When synchronizing tasks, it can be selected that just open tasks are exchanged. Furthermore, it can be filtered by partners, related to the task. When synchronizing contacts it can be chosen which type of contacts should be exchanged according to the partner organization.

The manual synchronization process is trigger if the user clicks the button "synchronize". If the users use quick synchronization then default or previous chosen settings are used. Before it is really synchronized, all contacts from Business One are read, then the

matching happens, conflicts are identified between contacts, and after that a synchronization preview window is displayed. This display shows which new data comes from which system and which conflicts occurred. The conflicts are opposed side by side and the user can chose which system should be trusted or can manually assign each field. To finish the synchronization the user has to accept the changes by clicking the button "synchronize all". The synchronization process can also happen automatically by making a schedule. Such a schedule can define if the synchronization process is triggered if Outlook is closed or opened, after a specific time, or if Outlook is logged on or off to Business One. However, to make it work, Outlook has to be running.

3.2. Microsoft Dynamics AX Outlook integration

Microsoft Dynamics AX is an ERP system for medium sized and large enterprises. It a competitor to SAP Business One and thus also to Semiramis. The ERP system is not only based on the visual design of Outlook, it also offers a support for its integration. The main functionality is the synchronization of contacts, appointments, tasks, and activities.

Instead of the SAP Business One solution, this one is an ERP sided implementation. That means all the synchronization activities are performed in Dynamics AX and not in Outlook. There is no need to install an Outlook add-in. The set up for the integration takes place in the ERP system. The folders where the tasks, contacts and calenders in Outlook are stored can to be directly selected. The user is identified by its Email address, so it has to be properly set in Dynamics AX and Outlook.

In Dynamics AX in the customer view, it is possible to export a contact to Outlook with a click on the Dynamics AX tool bar button "Add to Outlook Contacts". Furthermore, it is possible to add contacts from Microsoft Office Outlook when adding a new contact in Dynamics AX. There is also a option for a total synchronization. Here it can be chosen what should be synchronized and the time frame of the appointments can be selected. When the synchronization is done an info dialog shows the number of contacts, tasks and appointments synchronized. Special conflict resolution options are not available.

3.3. Sage CRM Outlook Integration

Behind Oracle and SAP, Sage is the world's third-largest supplier of ERP systems. Sage CRM is an addition to the Sages ERP systems, but can also operate as standalone application. Sages customer relation management solution Sage CRM has pre-installed Outlook integration capabilities. The main functionalities are the synchronization with Outlook contacts, calendar, and tasks. Furthermore, Emails can be attached to customers in Sage CRM. Additionally, it is possible to send Emails within Sage CRM using Outlook and automatically have Emails, including attachments, stored in the relevant customer contact history.

The installation is based on an Outlook plug in. In Outlook it has to be connected to Sage CRM. It is quite similar to the solution of SAP's Business One. Therefore, also additional buttons are added to the Outlook tool bar in order to use the integration functionalities. However they big difference is that the solution is not only a data integration, it is also a presentation layer integration. That means Sage CRM can be embedded into Outlook as a new view.

The most important new buttons are "Synch", "Add contact" and 'File Email". In the tool bar the "Add contact" button allows the user to store a single Outlook contact into the Sage CRM. It opens a pre-filled Sage CRM form embedded in Outlook where the user can decided to make changes and store it. The button "File Email" attaches the chosen Outlook Email automatically to the corresponding customer in Sage CRM, determined by the Email address. Here also a Sage CRM view is opened, which allows the user to manually chose where to attach the Email, for example, to a company, case or opportunity. The button "Synch" automatically synchronized the two systems. The synchronization settings are defined by CRM administrators.

CRM administrators can set the direction in which is synchronized: bi-directional or just in one direction. For conflict resolution it is possible to chose which system should be dominant. The administrators can also determine the schedule in which intervals it will be automatically synchronized.

4. Requirements analysis

In this chapter the performed requirement analysis procedure and its results are explained. As mentioned before, design science research answers question relevant to human problems. So, before we can start, we have to define that problem and thus, the research question. To that end, a requirement analysis is very helpful, not only for identifying and recording the actual problem at hand, also for defining measurements and benchmarks for its solution.

The requirement analysis procedure conducted is structured into project constraints, system constraints, functional requirements, non-functional requirements, operational scenarios and design problems. In order to define the project constraints, the first ideas were worked out by a meeting with a Comarch team. These baselines were reviewed and refined by the research and development (R&D). In the next step existing approaches for PIM integration and market analyses were involved to define external constraints. On the basis of the previous steps a questionnaire was developed and conducted to learn about the customer expectations. Following that, the system constraints were draw up. Here the system boundaries, interfaces, utilization environments, and project management were specified. Since the system's interfaces are significant for this project, certain PIMS were examined in order to learn which interfaces are the most common. At this point all problems, challenges, as well as opportunities were collected. In the next step functional and non-functional requirements were defined to summarize the essential findings. Following that, operational scenarios were developed, in order to specify how the functional requirements should be met. Finally, the general design problems representing the research question were stated.

4.1. Project Constraints

In this section the project constraints are depicted, which outline the steps from single ideas to a self-contained project.

4.1.1. Project and Enterprise Constraints

The idea of integrating PIMS and Semiramis arose from Semiramis customers asking for a solution in the case of exchange partner and customer data between the PIMS Microsoft Outlook and Semiramis.

The project was first elaborated by a meeting of us, with the CEO and a senior consultant of Comarch Solutions GmbH Austria, and an University of Innsbruck employee working in the Semiramis Research and Service Unit, and at the same time representative of the project Comarch Innovation Lab. During the meeting a few specifications and baselines were developed:

- Microsoft Outlook interface for exchange of appointments and contacts.
- Possibility to archive Outlook emails in Semiramis.
- The highest priority was determined for the synchronization of contacts, second appointments and activities, and lowest priority was assigned for the archiving of Outlook emails in Semiramis.
- Semiramis should be the leading system, that means that all functionalities should be as Semiramis centered managed and controlled as possible.
- Only for the user authorized data should be addressed by these functionalities.

After the meeting the R&D department and the product management department of Semiramis was contacted, in order to learn about their requirements in relation to our planned project. They expressed interest of:

- Neutral integration without fixation on Microsoft Outlook, other alternatives, like Mozilla Thunderbird, have to be considered too.
- Importance of conflict resolution options for the synchronization.
- At the synchronization of activities and tasks, it should be possible to include attachments and to conclude of tasks.
- Possibility to attach emails as documents to partners, orders, articles and other Semiramis objects.
- Optimally, stored emails should be tagged.
- Data synchronization should be either invoked manual or time-controlled.
- Automatized jumping from a Semiramis date field and from Semiramis workflow tasks directly into the PIMS's calendar.

The enterprise and Semiramis general specifications, standards, or guidelines are specified by the Semiramis style guide handbook. All standard applications in Semiramis have a embedded help to support the user. That means there is a help function which links the user a corresponding help document. In order to seamless integrate into the Semiramis application landscape, also the PIM synchronization application has to provide this feature.

Since this master project is a one-man and complimentary student work, definitions in regard of financial and human resource allocations, and team assignments and structure can be omitted at this point.

4.1.2. External Constraints

The project has not only to met enterprise standards, also public and international laws and regulations have to be followed. Optimally, the project complies to industry, international, and other general specifications, standards, and guidelines.

An important role is the market share of PIMS in order to determine which PIMS should be supported with the highest priority. Unfortunately there are no studies on the market share of PIMS, however there are several for email clients. Since in nearly any case an email client software has PIM functionality, these studies are sufficient in this matter. The email marketing software company Campaign Monitor has measured email client popularity through customers using their software (campaignmonitor.com, 2011). The study has started in 2009 and has been published on May 2011. The most popular tool was Outlook with a total share of 27.63%, which is comprised of Outlook 2000, 2003 and Express with 17.88% , and Outlook 2007 and 2010 with 9.74%. The following results were: iOS Devices (iPhone, iPad and iPod Touch) with 16.01%, Hotmail with 12.14%, Apple Mail with 11.13%, Yahoo! Mail with 9.54%, Gmail with 7.02%, Windows Mail with 1.84%, Android with 1.70%, AOL with 1.25%, Thunderbird with 1.21% and Lotus Notes with 0.20%.

A similar study was conducted by Litmus which is also a software company for Email analytics (litmus.com, 2010). Here the Email client usage worldwide, collected from 250 million users, was analyzed and published on February 2010. The most popular one was also Outlook with a total share of 43%, which is comprised of Outlook 2003, and earlier with 34%, and Outlook 2007 with 9%. The other result were: Hotmail with 17%, Yahoo! Mail with 13%, Gmail with 5%, Apple Mail with 4%, iPhone with 4%, Thunderbird with

2.4%, Windows Live Mail (Desktop) with 2%, AOL Mail with 1.2%, Lotus Notes with 0.4% and others with 8%.

The clear result is that Outlook has the highest priority followed by the Apple's products the mobile iOS Devices and Apple Mail, and webmail clients Hotmail, Yahoo! mail and Gmail.

Also important regarding the external constraints is the definition of the technology base. The chosen technologies are described in more detail in Section 5.1. The main technology base is defined by Semiramis, since it is the leading system and will contain most of the functionalities. However the technologies of the chosen PIMSs are playing a big role too.

Influencing the external constraints are also the capabilities of competitor products. These softwares were examined in more detail in Chapter 3. The analysis of competitor products has shown that many functionalities are already covered by the requirements determined before. However, especially when it comes to conflict resolution, some strategies are wide spread among PIMS integration solutions. Thus these have to be considered in the requirements. Manual conflict resolution is often applied with the help of a synchronization preview. Before the data is synchronized, the user can view the changes and conflicts, which have encountered. Here the user is able to interfere. Automatic conflict resolution is often handled by choosing a dominant system. The data on the dominant system is then considered as the correct data and will overwrite the data from the other system. Furthermore competitor products have often the option to chose between the directions. The synchronization can be either in one direction or in both.

4.1.3. Customer Expectations

For this master project no specific customer can be assigned, because no specific external organization has contracted the project. However, there exist several potential Semiramis customers which have interest in this application. Therefore, in order to gather representatively customer expectations a questionnaire was designed. The questionnaire and further details can be found in the appendix section A. It was designed to learn as much as possible about the participant PIM habits and their personal attitude towards data quality related to it. However, the main emphasis of the questionnaire was to collect the participants opinions on the possible functionalities of the synchronization application.

The PIMS used by the participants were: MS Outlook with 91%, Outlook Web Access with 33%, Google Mail with 25%, Mozilla Thunderbird with 8%, Tobit with 8% and Lotus Notes with 8%. Since mutliple answers were allowed the percentage exceeds 100%. Outlook Web Access or also called Outlook Web App is the webbased webmail service for MS Exchange servers. All of the partakers used their PIMS for sending and receiving emails, and for contact management. 92% use them for time mangement, 33% for document and memo management, 17% use them for knowledge management, customer relationship management or news subscription. 75% use their PIMS for the synchronization with mobile devices and 25% with other systems. 92% of the participants had one or more company mobile devices capable of PIM.

The questionnaire also asked the participants about the importance of possible functionalities of the synchronization application. The results are summarized in Figure 4.1. The highest interest had partner synchronization with 3.15 out of 4 points, followed by document exchange with 3 points, email import with 2.77 points, and appointments synchronization with 2.54 points. There was more interest in manual synchronization, 2.92 points, than in time triggered, 2.41 points. The most preferred synchronization direction was voted for bidirectional, 2.92 points, followed by unidirectional from Semiramis to a PIMS, 2.77 points, and unidirectional from a PIMS to Semiramis, 2.31 points.

From the technical point of view, the most, with 38%, wished for synchronization on server-level between Semiramis and a MS Exchange server. 23% would prefer a middleware and 15% a synchronization on the client level.

To sum it up the results mirror quite similar the major findings of the previous two subsections. It again showed that MS Outlook is the most widespread PIMS, and partner synchronization has the highest priority regarding functionality. However it also shows that customers consider document exchange and Email import more important than appointments synchronization. Especially, it pointed out the importance of mobile devices in context of personal information synchronization. The results also exposed the wish for synchronization with MS Exchange servers. All in all no redundant functionalities, which have, for instance, no use for the customer, were observed.

4.2. System Constraints

This section examines the system constraints, these determine the system's boundaries, interfaces, utilization environments and project management.

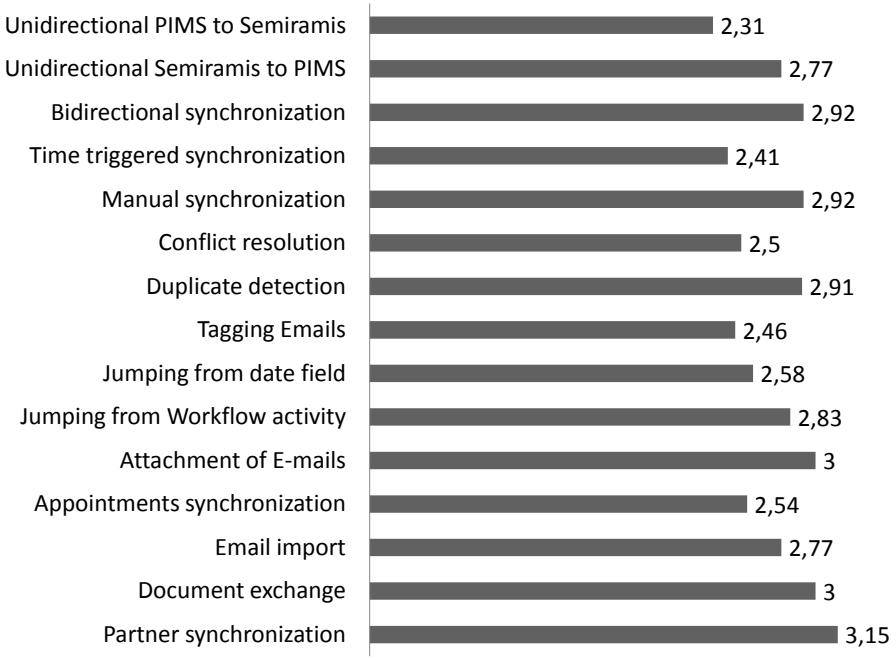


Figure 4.1.: Interest of customers by functionality, where the maximum is 4 points.

4.2.1. System Boundaries

Here it is important to define which system elements are under design control of the developers and which fall outside.

The applications underlying system Semiramis is just to a certain degree under design control of us. Under design control is the background logic of an application. However the visual appearance itself, the look and feel, is not. This ensures that Semiramis appearance is always consistent. The arrangement of certain GUI elements, inside an application, are in hand of the developer. But, the Semiramis style guide standardizes this. The expected interactions between system elements outside and inside the system boundaries takes place via interfaces and data containers files. These are dealt with in the next subsection.

4.2.2. Interfaces

Here the functional and design interfaces to external elements, like systems, platforms, humans, are defined.

The interfaces provided to humans is completely supported by Semiramis. Standard Semiramis application presentations will be used. The advantage of this is that Semiramis design standards a complied and users are familiar with the interaction possibilities. A Semiramis customizable cockpit application will be used to interact with the user. A customizable cockpit is a dialog application which consist of two parts: a search mask and a table. The search mask allows to enter several search parameters. The table represents the results of this search. A cockpit allows the manipulation of several information objects through their selection in the table. Actions to interact with the chosen records can be found in the Semiramis tool bar. Furthermore, the Semiramis customizing application will be used to configurate the synchronization application which may run in the background, for instance, if it is time triggered. The customizing application is a special dialog were Semiramis users can select specific parameters for various applications and background tasks.

Before we can define interfaces to other systems and platforms, we have to analyze which protocols and data formats are best supported by the most relevant PIMS. There are several data formats and protocols suited for exchanging PIM data, the most important ones are shortly mentioned in the following. SyncML is a platform-independent information synchronization standard which is also describing a protocol for the exchange of arbitrary data. In most cases SyncML is used for the synchronization of PIM data across computers and mobile devices. The protocol CalDAV is an extension to the HTTP-based protocol WebDAV, with the special purpose to access scheduling information on a remote server. LDAP is a protocol for accessing directory information services, for PIM purposes it is mainly used to access contact information. LDIF is data interchange format for representing LDAP directory content. While iCalendar is a standard format for representing calendar information, vCard is designed to represent contact information. CSV and Personal Folder Files are multi purpose data file formats which can contain all kinds of personal information. The relevant protocols and data formats will be explained in more details in Chapter 5.1.

The table 4.1 shows which data formats and protocols for the exchange of PIM data are natively supported by selected PIMS solutions.

Table 4.1.: Native supported data and protocol interfaces by PIMS. Supported standard Email protocols, like POP3, SMTP and IMAP, are not listed here.

PIMS	**Import Data Formats**	**Export Data Formats**	**Supported Protocols**
MS Outlook	Personal Folder Files, iCalendar, vCard, Personal Address Book, CSV, Lotus Organizer, ACT!, MS Access Database, MS Excel	Personal Folder Files, MS Access Database, MS Excel, CSV, iCalendar, vCard	LDAP
Mozilla Thunderbird with Lightning	CSV, LDIF, vCard, iCalendar	CVS, LDIF, vCard, iCalendar	LDAP, CalDAV
Google Mail	CVS, iCalendar, vCard	CVS, iCalendar, vCard	CalDAV
Lotus Notes	CVS, iCalendar, vCard, Lotus 1-2-3	CVS, iCalendar, vCard, Lotus 1-2-3	LDAP

From the table we can see that every listed PIMS supports the file formats: iCalendar and vCard. Consequently, supporting these formats has the highest priority. There is also an urgent need for CVS support, because the most wide spread PIMS MS Office is not able to import and export vCards with multiple entries. On the other hand when it comes to protocols such equally clear support of specific ones cannot be recognized. However, there exist numerous extensions, workarounds and services for each PIMS application in order to support additional file formats and protocols. An extensive investigation on synchronization protocols has resulted that SyncML is one of the most common platform independent personal information synchronization standards. There are SyncML plug-ins available for many PIMSs, and furthermore, it is also widely supported among mobile phones. Therefore for automatized synchronization SyncML was chosen.

4.2.3. Utilization Environments

Here the environments the system operates and corresponding environmental factors are defined, which may propose effects on hardware, software, and humans and thus impact system performance. Environmental factors may emerge from daytime, weather conditions, geographical topology, biology and other induced effects.

In our case the system operates in a normal organizational office environment. Therefore, no factors regarding the environment can be identified which may affect the system's performance or harm the user in an unexpected way. However, it can be assumed that at night times and outside office hours of operation, is suited best for synchronizing large data sets, since the system is fewer utilized.

4.2.4. Project Management

Here we define system's life cycle process requirements with emphasis on humans involved. The system's project management includes the tasks: development, production, testing, distribution, operation, support, training, and disposal of the system and it's products. We have to determine the manpower, training, competences and human engineering needed for each task.

As mentioned before, for the development and production no additional manpower is needed since it is done during the master's thesis. The application should be documented and designed in a way that no specific manpower is needed for the support, disposal and training. For the operation and distribution the already existing Semiramis infrastructure will be used, thus no additional manpower will be needed.

The distribution and disposal of the application Semiramis administration experiences will be required. The competences needed for the testing and operation of the application are basic Semiramis user experiences. The competences required for the development, production, and support of the system are corresponding to the skill set of a Semiramis application developer.

Special additionally training or human engineering should not be needed since the project complies with Semiramis standards and a Semiramis integrated manual will be provided.

4.3. Functional Requirements

The functional requirements determine what the application should be able to do. For example, functional requirements can be technical details, data manipulation, calculations and processing functionality.

The complete list of functional requirements which were derived by the previous sections can be seen in Table 4.2

Table 4.2.: Functional requirements. D stands for dependencies and P for priority.

#	**Description**	**Source**	**D**	**P**
R1	Implementation of a PIMS interface for exchange of contacts with Semiramis.	Meeting		high
R2	Implementation of a PIMS interface for exchange of appointments, tasks and activities with Semiramis.	Meeting		medium
R3	Implementation of a PIMS interface for storage of emails in Semiramis.	Meeting		medium
R4	Include attachments of exchanged activities and tasks.	Product Management	R2	medium
R5	Synchronization of the conclusion of tasks.	Product Management	R2	medium
R6	Attachment of emails in Semiramis as documents to several business objects.	Product Management	R3	low
R7	Tagging of imported emails in Semiramis.	Product Management	R3	low
R8	Data synchronization is invoked manual or time-controlled.	Product Management		high
R9	Jumping from a Semiramis date field into the PIMS's calendar.	Product Management	R2	low
R10	Jumping from a Semiramis workflow activity into the PIMS's calendar.	Product Management	R2	low
R11	Automatic conflict resolution by choosing a conflict resolution option.	Competitor product analysis		medium
R12	Partner are be exported in formats: CVS and vCard.	Interface analysis	R1	high
R13	Partner are be imported in formats: CVS and vCard.	Interface analysis	R1	high
R14	Synchronization via SyncML.	Interface analysis		high

The first column contains the identifiers for the requirements, for example the use cases will reference to these. The column with the title 'source' indicates where the idea

originated. The column 'D' point out if a requirement depends on another requirement. For example requirement R7 is dependent on requirement R3, because the tagging of imported emails is only possible if the functionality of storing email in Semiramis is already implemented. The column 'P' states the priority of the requirement. The priority is mainly influenced by the questionnaire's answers, but also by the opinion of the Comarch team.

4.4. Non-Functional Requirements

Non-functional requirements are additional requirements which do not contribute to the functionality itself, but have to be covered as well, since they are describing the softwares characteristics. Thus, not how the system should be supposed to do, but to be. It can be said that the non-functional requirements are the qualities of a product. The software engineering standard ISO/IEC 9126 for product quality classifies software quality in the main characteristics: Functionality, reliability, usability, efficiency, maintainability, and portability (ISO/IEC, 2001). Functionality describes to which extend the functional requirements are implemented. Sub-characteristics of functionality are: Suitability, accuracy, interoperability, security, and functionality compliance. On the other hand efficiency or sometimes referred to performance requirements defines how well the functional requirements should perform. Sub-characteristics belonging to efficiency are: Time behavior, resource utilization, and efficiency compliance. The characteristic reliability describes the ability to maintain its performance niveau under certain circumstances. Sub-characteristics of reliability are: Maturity, fault tolerance, recoverability, and reliability compliance. Usability describes the ease of use and how the usability is assessed by individuals. Sub-characteristics are: Understandability, learnability, operability, attractiveness, and usability compliance. Maintainability qualifies the effort needed for making changes and corrections to the software. Sub-characteristics of maintainability are: Analyzability, changeability, stability, testability, and maintainability compliance. Portability depicts the possibilities and effort to transfer the product to other hardware, software, or organizational environments. The portability's sub-characteristics are: adaptability, installability, co-existence, replaceability, and portability compliance. In Table 4.3 the non-functional requirements and corresponding characteristic type are listed.

Table 4.3.: Non-functional requirements. P stands for priority. Char stands for characteristic.

#	**Description**	**Source**	**P**	**Char (Sub-Char)**
NR1	Semiramis as the leading system.	Meeting	high	Functionality (Compliance)
NR2	Only access to authorized data.	Meeting	high	Functionality (Security)
NR3	Neutral integration without fixation on a specific PIMS.	Product Management	high	Portability (Adaptability)
NR4	Compliance with Semiramis style guide.		high	Functionality (Compliance)
NR5	Compliance with Comarch guidelines, procedures and policies.		high	Functionality (Compliance)
NR6	Compliance with public and international laws and regulations.		high	Functionality (Compliance)
NR7	Compliance with industry, international, and other general specifications, standards, and guidelines.		high	Functionality (Compliance)
NR8	Supporting PIMS: MS Outlook.	Market Study	high	Functionality (Interoperability)
NR9	Supporting PIMS: iOS Devices and Apple Mail.	Market Study	medium	Functionality (Interoperability)
NR10	Supporting web-based PIMS: Hotmail, Yahoo! mail and Gmail.	Market Study	low	Functionality (Interoperability)
NR11	Technology base: Semiramis and chosen PIMS.		high	Functionality (Compliance)
NR12	Supporting mobile PIMS.	Questionnaire	high	Functionality (Interoperability)
NR13	Supporting MS Exchange server.	Questionnaire	medium	Functionality (Interoperability)
NR14	A Semiramis integrated user manual.	Meeting	high	Usability (Understandability)
NR15	Errors should be caught and dealt with meaningful.		high	Reliability (Fault Tolerance)

Table 4.3.: Non-functional requirements. P stands for priority. Char stands for characteristic.

NR16	Customizing options should support the duplication detections decisions.		high	Functionality (Suitability)
NR17	Customizing options should support the connecting to the SyncML server.		high	Functionality (Suitability)
NR18	A secure connection should be used.		high	Functionality (Security)
NR19	The synchronization process should work without user intervention.		high	Reliability (Maturity)
NR20	No additional training costs should arise.		high	Usability (Learnability)
NR21	Semiramis version 5 is needed.		high	Portability (Installability)
NR22	Easy to install through Semiramis.		high	Portability (Installability)
NR23	Synchronization process should cost as few resources as possible.		high	Efficiency (Resource Utilization)
NR24	Synchronization process should run in background so that running time is not time costing for users.		high	Efficiency (Resource Utilization)
NR25	No duplicates and redundant data should be generated.		high	Functionality (Accuracy)
NR26	The data format exported should be valid to the standards.		high	Functionality (Compliance)
NR27	The newly data to be stored should be valid to Semiramis standards.		high	Functionality (Accuracy)
NR28	The duplicate detection should work correctly and as error free as possible.		high	Functionality (Accuracy)
NR29	The SyncML server should be exchangeable.		high	Portability (Adaptability)

Table 4.3.: Non-functional requirements. P stands for priority. Char stands for characteristic.

NR30	The SyncML implementation should be conform with SyncML standard.		high	Functionality (Compliance)
NR31	Available conflict options: Semiramis wins, conflicts are shown to the user, and conflicts are not stored.		high	Functionality (Suitability)
NR32	Partner attributes to be exported: Person data (name, title, etc.), employment relationship, address data, and communication data.		high	Functionality (Suitability)

4.4.1. Modes of Operations

The application has just one mode of operation, which is fully operational. In this mode all functionalities are available.

4.4.2. Design Characteristics

The design characteristics are solely based on the Semiramis standard. That means consistent GUI design.

4.4.3. Human Factors

Human-factor have to be considered by consider ergonomics, cognitive limits, and usability. This factors are already covered by the Semiramis standard.

4.5. Operational Scenarios

In this subsection we define the range of the anticipated uses of the application, and its services and products. Furthermore, the expected interactions with the environment and other systems are determined. Thus, human tasks and task sequences, and physical

intercommunication with interfacing systems, platforms, or products are defined. In order to define operational scenarios we can describe user scenarios and use cases (compare Subsection 2.5.1).

At first we have to define the actors. The most important one is the user, which wants to synchronize his data. In our case it is always a human or a group of humans. A further actor is the system Semiramis. On the one side it stores PIM data and on the other side it controls the synchronization by the help of user interaction. Then another actor is a system which can be an arbitrary PIMS, which also stores PIM data.

The products which are produced by the application are data set files containing PIM data in a standardized format.

Because of the limited scope of this master project, not all functionalities will be covered with the first version of the synchronization application. Since the synchronization of contact data has the highest priority, this feature depicts the minimum functionality. The scenario covered are shown in Table 4.5, 4.6 and 4.7. The primary function implemented by the first prototype is partner synchronization which is achieved with the help of the SyncML standard and thus via a SyncML server. For later work the synchronization with a SyncML server can be easily extended in order to include further functions like appointment and document synchronization. Secondary the PIM synchronization application is also supporting manual contact data exchange between a PIMS and Semiramis through partner import and export functions. The import and export functions will support the file format CSV and vCard.

The tables describe the use cases, including scenarios, of synchronize PIM and contact data, and exporting and importing contact data.

Table 4.4.: Use case 1: Synchronize PIM data.

Use Case Name	**Synchronize PIM data**
Traceability Identifier	UC-1
Description	The process of continuous or frequently alignment of personal information between Semiramis and a PIMS.
Actors	User, Semiramis, PIMS
Performance Goals	To have up to date PIM data available to both Semiramis and PIM.
Preconditions	Application is configured properly by the user. There is PIM data to synchronize.
Postconditions	Data is consistent.
Scenario	**Begins** when PIMS or Semiramis notice that data has changed or when a preconfigured time has passed. 1. PIM data gets validated by Semiramis, for example, if its exchange is authorized. 2. PIM data is converted by Semiramis into a standardized exchange format. 3. Secure connection is established by Semiramis. 4. PIM data is send to the other end by Semiramis. 5. PIM data gets validated at the other end. **Ends** when data has been synchronized and is persistent.
Alternative Scenario	ad 1 and 5. Validation fails: Process is aborted **Ends** when error message is thrown by Semiramis. ad 3. Connection fails: Process is aborted **Ends** when error message is thrown by Semiramis.
Functional Requirements Covered	R1, R2, R3, R5, R8, R11, R14

Table 4.5.: Use case 2: Synchronize contact data.

Use Case Name	**Synchronize contact data**
Traceability Identifier	UC-2
Description	The process of continuous or frequently alignment of personal contact information between Semiramis and a PIMS.
Actors	User, Semiramis, PIMS
Performance Goals	To have PIM contact and partner data available to Semiramis and PIM
Preconditions	Application is configured properly by the user. There is PIM contact data to synchronize.
Postconditions	Partner data is consistent.
Scenario	Same as in UC-1.
Alternative Scenario	Same as in UC-1.
Functional Requirements Covered	R1, R8, R11, R14

Table 4.6.: Use case 3: Export contact data.

Use Case Name	**Export contact data**
Traceability Identifier	UC-3
Description	The process of manual and user initiated export of Semiramis partner data into standardized data formats.
Actors	User, Semiramis
Performance Goals	To have PIM contact and partner data available in standardized data formats so that various PIMS can use them further.
Preconditions	There is PIM contact data to export.
Postconditions	Partner data is exported into a valid data format.
Scenario	**Begins** when the user starts the Semiramis application. 1. User choses search parameter for contacts to export. 3. User submits to start search. 4. Semiramis searches for partners authorized to export. 5. Set of found partner are shown in a list. 6. User choses either all results or selects specific ones. 7. User selects the action to export partners. 8. Dialog opens where user choses data format and target file directory. 9. User submits to export. 9. Semiramis converts partner data into the chosen format. 10. Semiramis stores the output file. **Ends** when output file is persistent stored on a file system.
Alternative Scenario	ad 10: Semiramis fails to store file. **Ends** when error message is thrown.
Functional Requirements Covered	R11, R12

Table 4.7.: Use case 4: Import contact data.

Use Case Name	**Import contact data**
Traceability Identifier	UC-4
Description	The process of manual and user initiated import of Semiramis partner data from standardized data formats.
Actors	User, Semiramis
Performance Goals	To have PIM contact and partner data from various PIMSs which can be used further in Semiramis.
Preconditions	There is contact data available in a standardized data format.
Postconditions	Partner data stored is valid and persistent.
Scenario	**Begins** when the user starts the Semiramis application. 1. User selects the action to import partners. 2. Dialog opens where user choses target file. 3. User submits to import file. 4. Semiramis validates file format. 5. Semiramis processes the file. 6. Semiramis validates the resulting partners. 7. Semiramis checks for duplicates and conflicts. 8. Semiramis stores new partner entries. Data with conflicts are not stored. 9. Semiramis displays imported partners and throws error message showing the conflicts. **Ends** when user closes application.
Alternative Scenario	ad 4: File format is not correct. **Ends** when error messages are thrown. ad 6: Semiramis fails to store partners due to faulty data. **Ends** when error or warning messages are thrown.
Functional Requirements Covered	R11, R13

The below Figure 4.2 shows the use case diagrams for the use cases. It shows the relationships between the use cases, which are represented by ellipses. The rectangle denotes the borders of the PIM synchronization application. The actors are illustrated by stick persons.

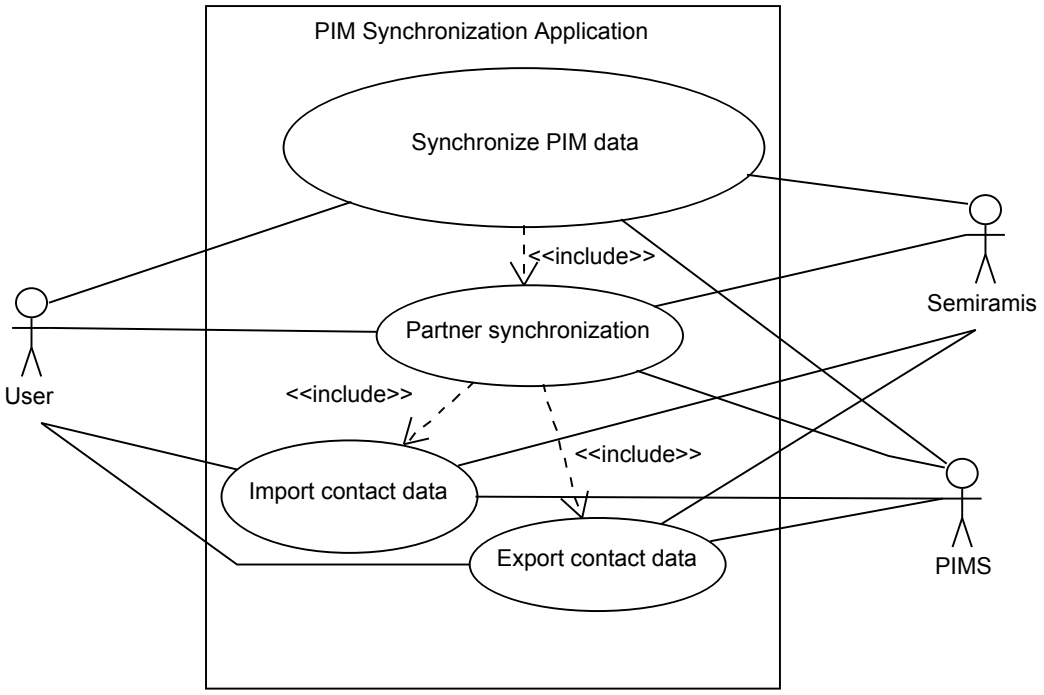


Figure 4.2.: Use case diagram for UC-1, UC-2, UC-3, and UC-4.

4.6. Design Problems

The design problems can be seen as the super problems which have to be overcome in order to satisfy the requirements. In this case, it is the main research question of this master thesis. The problem is described in the following.

On the one hand we have the ERP system Semiramis, running on an arbitrary server. On the other hand we have an arbitrary PIMS which is running on any computer or mobile device. There are no common interfaces yet. The questions are:

- Is it even possible to interchange PIM data between those systems?
- How can it be achieved?

- How has the data to be altered, in order to make them interchangeable?
- Which protocols and representation formats are suited for that task?
- How should the communication between those systems proceed?
- Which existing functionalities can be reused?
- What is the most practical way to select data to be exchanged?
- How do we prevent duplicate data entries?

In the next chapter these problems are tackled by designing a solution for the prototype.

5. Design

In this chapter the basis technologies, deployment, and architecture of the implemented prototype are described.

5.1. Basis Technologies

In order to support as many PIMS as possible it is advantageous to use open and wide spread standards for communication protocols and data formats. In the requirement analysis, especially in Subsection 4.2.2, we already examined which data formats and protocols fit best to the problem at hand. We decided to use the vCard data format, the CSV data format, and the SyncML synchronization standard. In the next subsections these basis technologies are explained in more details.

5.1.1. Semiramis

As already mentioned before, Semiramis is implemented in the object-oriented programming language Java. Therefore, also the development in Semiramis is done in Java. As software development environment the integrated development environment (IDE) Eclipse is used. IDE means that Eclipse has a built in source code editor, compiler and debugger. Furthermore, there exists a Eclipse plug-in for Semiramis integration. This plug-in, for example, allows the automatized creation of development projects out of Semiramis development jobs. A Semiramis development job is usually a collection of development objects. Development objects, for instance, are: Java files, XML definitions, database objects, or data descriptions. At a higher level, there is a Semiramis development task which again may consist of multiple development jobs. A development project in a properly configured Eclipse environment can be executed, in order to start a local Semiramis instance. The local Semiramis instance has integrated all changes made by the development job and can be debugged by the Eclipse integrated debugger.

The main objects the prototype will work with are contacts. Contacts are defined and stored in Semiramis as instances of the business object (BO) Partner. A BO is the representation of a database table and may have references to other business objects. The attributes of the business object Partner are for instance: number, type, name, and person. The person attribute is a complex attribute and consists further more of given name, middle name, surname, title, salutation, gender, and full name. The BO Partner has a reference to the BOs PartnerAddressData and CommunicationData. The Partner instances maintained by the Semiramis partner maintenance application, which is a graphical editor for creating, storing, updating, loading and duplicating partners. Semiramis is already capable of exporting partners as spreadsheet or XML files. Import is possible only by XML files. However, PIM software are not able to process these XML files.

5.1.2. vCard Data Format

A wide spread standard for storing contact information is the vCard data format (Consortium, 1997). It is a container for electronic business cards with the emphasis of exchange over the Internet. A vCard is able to store name and address information, phone numbers, e-mail addresses, URLs, logos, photographs, and audio clips. It is supported by many E-mail clients, PDAs, mobile phones and other PIM applications. The format was created by the Versit Consortium. Versit Consortium was founded by the organizations Apple, AT&T, IBM and Siemens in order to develop common standards for personal data interchange (PDI). Currently, the rights to the data format belong to the Internet Mail Consortium, which is a provider of information and developer of technologies related to Internet mail standards and PDI. The vCard is currently further developed by the Internet Engineering Task Force (IETF). The IETF is an organization with the aim to develop and promote Internet standards. The vCard version 2.1 is specified by the Versit Consortium in (Consortium, 1996), version 3.0 is specified by the IETF published standard RFC 2426 (Dawson and Howes, 1998). Version 2.1 is the most spread one. However, many applications support both.

A vCard is saved in a simple unformatted ASCII encoded text file, while the filename extension are commonly .vcf or .vcard. A vCard entry is donated within the opening tag BEGIN:VCARD and closing tag END:VCARD. The following tag VERSION indicates which specification version of vCards is used. The properties and attributes of a contact

information are of the format: PROPERTY[;PARAMETER]:Attribute[;Attribute]. One vCard file can contain multiple vCard entries. An example can be seen in Listing 5.1.

Listing 5.1: vCard version 2.1 example.

```
BEGIN:VCARD
VERSION:2.1
N:Musterfrau;Helga;;Dr.;Frau;
FN:Dr. Helga Musterfrau
ADR;HOME:;;Stationstreet 2;New York City;;10001;USA
URL;WORK:www.example.com
EMAIL;INTERNET;WORK:musertfrau.helga@example.com
TEL;VOICE;WORK:55666111
END:VCARD
```

5.1.3. CSV Data Format

The acronym CSV stands for comma separated values. It is a multiple purpose plain text file format for storing information tabular. Columns/fields are separated by comma or semicolon characters, and rows/records by line breaks. In general, the first line is the header record which defines the field names. However CSV is not strictly standardized, for example when it comes to character encoding, comma character, date format, or newline symbol. In our case we refer to the standard RFC 4180 (Shafranovich, 2005). It uses the normal comma character ',', the line break type is CRLF, and each field may or may not be enclosed with double quotes. CRLF, carriage return and line feed, is used by the Windows operating system. As already mentioned before in the requirements analysis MS Office is the reason why this format has to be supported. Therefore the CSV file formats header record will be based on the MS Office preferred one. One problem though is, that the header record is language dependent. For example an English installation of MS Office is not able to automatically map the fields from a German installation. Therefore, it will be required to offer a multi language support. An example of an English CSV file can be seen in the below listing.

Listing 5.2: CSV Example with two contacts.

```
First Name,Last Name,Title,Suffix,Web Page,E-mail Address
Testmann,max,,,,example@org.de
Helga,Musterfrau,Dr.,Frau,www.example.com,musertfrau.helga@example.com
```

5.1.4. SyncML Synchronization Standard

The acronym SyncML stands for synchronization markup language. It is an open standard providing a platform independent information synchronization framework and protocol. It is commonly used to synchronize contact and calendar information, however it also able to deal with the exchange of other PIM data like memos, tasks, book-marks, Emails, SMSs, photos, videos, music or any other arbitrary files. The SnycML standard was developed by the standards organization Open Mobile Allicance (OMA) and is specified in the OMA SyncML Common Specification V1.2.2 (OMA, 2009). The specification contains the main components: XML-based representation protocol, synchronization and device management protocol, and transport binding for the protocol.

The SyncML framework is based on a client/server architecture. The communication takes place through SyncML XML messages, which are very similar to a Email message. Thus, they contain receiver and sender information, additional unique synchronization IDs for the sender, and a synchronization instruction. The synchronization instruction could be either an add, delete or update statement. The unique synchronization IDs identify data entries to be processed. The advantage of the use of XML is, that these messages are platform and implementation independent. The XML files can be exchanged via different transport protocols like OBEX, WML or HTTP. The synchronization process is usually triggered off by the client. An example XML can be seen in the below listing.

The main concepts which are dealt by the SyncML standard are: ID handling, change detection, modification exchange, conflict detection, conflict resolution, slow and fast synchronization. ID handling means that every data record is allocated a unique ID, in order to identify identical data sets. The IDs at the client can vary to the IDs at the server. In this case a mapping table is needed. Change detection determines if a data record has changed and when. Modification exchange defines how a modification takes place, for example if an entry has to be deleted, replaced or newly generated. Conflict detection is responsible for recognizing the above describe cases. Conflict resolution decides how a conflict is solved and which data should be taken as reference. Slow synchronization means that all data is synchronized, whereas fast synchronization makes sure that only changed data is exchanged.

During a normal session, the client connects with the server and asks for a synchronization. If the server agrees, both exchange their last synchronization anchors, if these differ a slow synchronization is started. Else, the client send only its modified data to

the server. The server deals with conflicts and re-adjust his data. Then the server send back its modified data so that the client can also update his data. If the client assigns new IDs to the data, it sends a mapping table to the server, which stores them.

Listing 5.3: SyncML XML Example: Server replies to client's request for authentication.

```
<SyncML>
  <SyncHdr>
    <VerDTD>1.0</VerDTD>
    <VerProto>SyncML/1.0</VerProto>
    <SessionID>1</SessionID>
    <MsgID>2</MsgID>
    <Target><LocURI>IMEI:123</LocURI></Target>
    <Source>
        <LocURI>http://www.example.org/sync-server</LocURI>
    </Source>
  </SyncHdr>
  <SyncBody>
    <Status>
      <MsgRef>1</MsgRef><CmdRef>0</CmdRef><Cmd>SyncHdr</Cmd>
      <TargetRef>http://www.example.org/sync-server</TargetRef>
      <SourceRef>IMEI:123</SourceRef>
      <Data>212</Data>
      <!--Alert code 212: Authentication successfull-->
    </Status>
    ...
  </SyncBody>
</SyncML>
```

5.2. Deployment

The deployment diagram of Figure 5.1 visualizes the topology of the application integration. The central system is the Semiramis application server which runs on an arbitrary server. On the Semiramis application server the implemented synchronization prototype is situated. The user can access Semiramis through Internet communication protocol HTTPS, through the Internet or an intranet, with the help of a browser. Through input

and output interfaces the prototype is able to import and export vCard and CSV files, which may origin and be processed from any PIMS. The communication between the prototype and a SyncML server takes place via XML streams, conform to the SyncML standard, through HTTP or HTTPS. The synchronization process is independent of the SyncML server and its location. Any SyncML client, regardless if situated on a mobile or desktop device, can communicate with the SyncML server through the supported protocols. Some SyncML servers also provide hosted services to enable direct connection and management of the data with a browser.

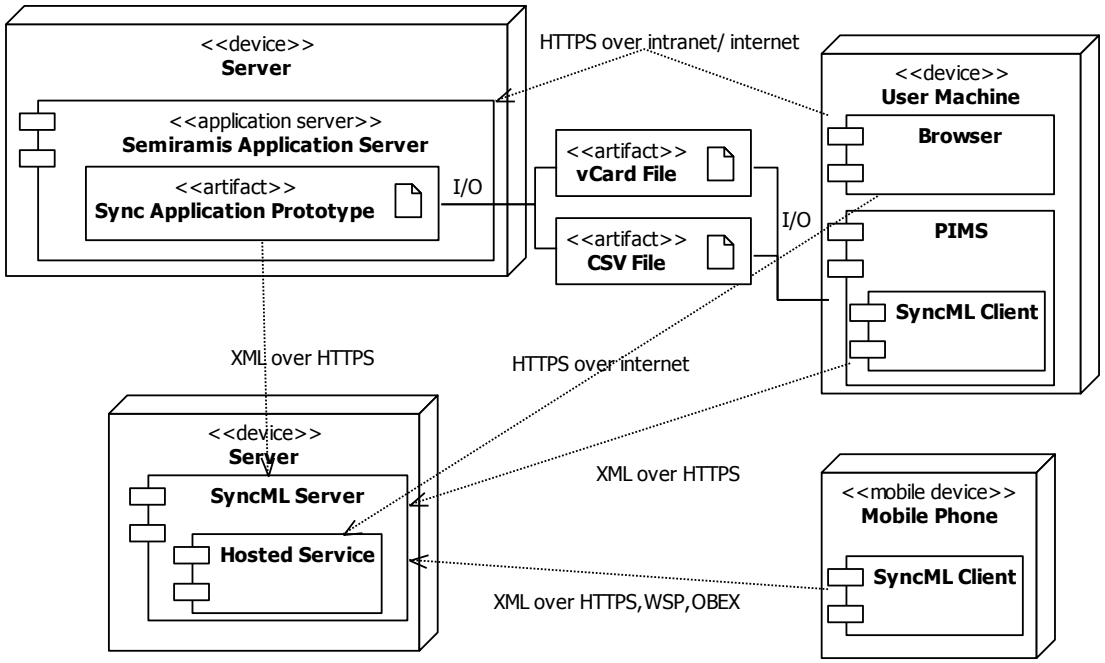


Figure 5.1.: UML deployment diagram of the PIM data integration.

5.3. Architecture

Figure 5.2 outlines the main applications which are three background applications ImportContacts, ExportContacts, and SyncWithSyncML, and the GUI application Partner-

Cockpit. All four classes extends the abstract Semiramis standard class CisApplication. CisApplication already implements most of the functionalities all Semiramis applications must have to allow their invocation. The background applications additionally extend the abstract Semiramis standard class CisBatchApplication. The CisBatchApplication provides further features needed by background applications. For example, their possibility for time triggered invoking or their possibility of displaying a GUI dialog on manual execution. This dialog allows the user to enter parameters which may be needed by the background applications. For time triggered invoking these parameters can be stored. The PartnerCockpit is the main visual application. It is a search application which allows the query for partners. Through the PartnerCockpit all three background jobs can be started instantly or time triggered. The background applications ExportContacts and SyncWithSyncML need as parameter a list of partners which should be exported. These list of partners is the search result, or a selection of it, coming from the PartnerCockpit's query.

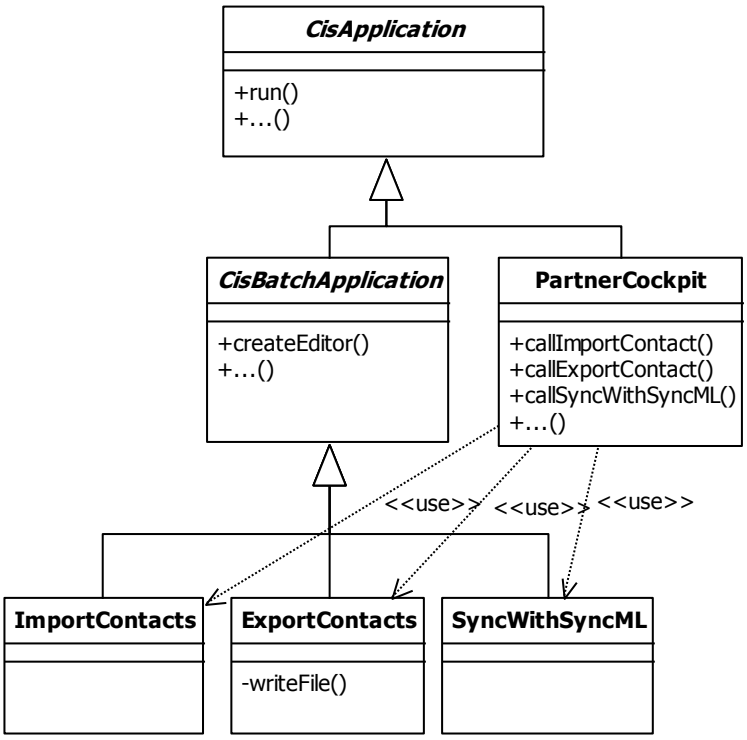


Figure 5.2.: UML class diagram of the main applications.

The two background applications ImportContacts and ExportContacts are described in more detail with the class diagram in Figure 5.3. Depending if it is dealt with a CSV or

vCard file, both applications use the VcardLogic and CsvContactLogic, in order to import or export partners. The classes VcardLogic and CsvContactLogic extend the abstract class AbstractContactLogic, which already implements equal functionalities. Both classes are able to exchange between a Semiramis partner and the corresponding CSV or vCard representation. To that end, they use the Semiramis standard class PartnerEntity. The PartnerEntity is a logic class which is also a container for the real Partner object. The PartnerEntity, for example, is available of instantiating, validating, authorization checking, and storing a partner. The Semiramis guidelines strictly advise against the direct processing of a partner object. The class Partner is a direct representation of a partner in the database. The VcardLogic and CsvContactLogic use the PartnerDuplicateLogic for duplicate detection at the import of partners.

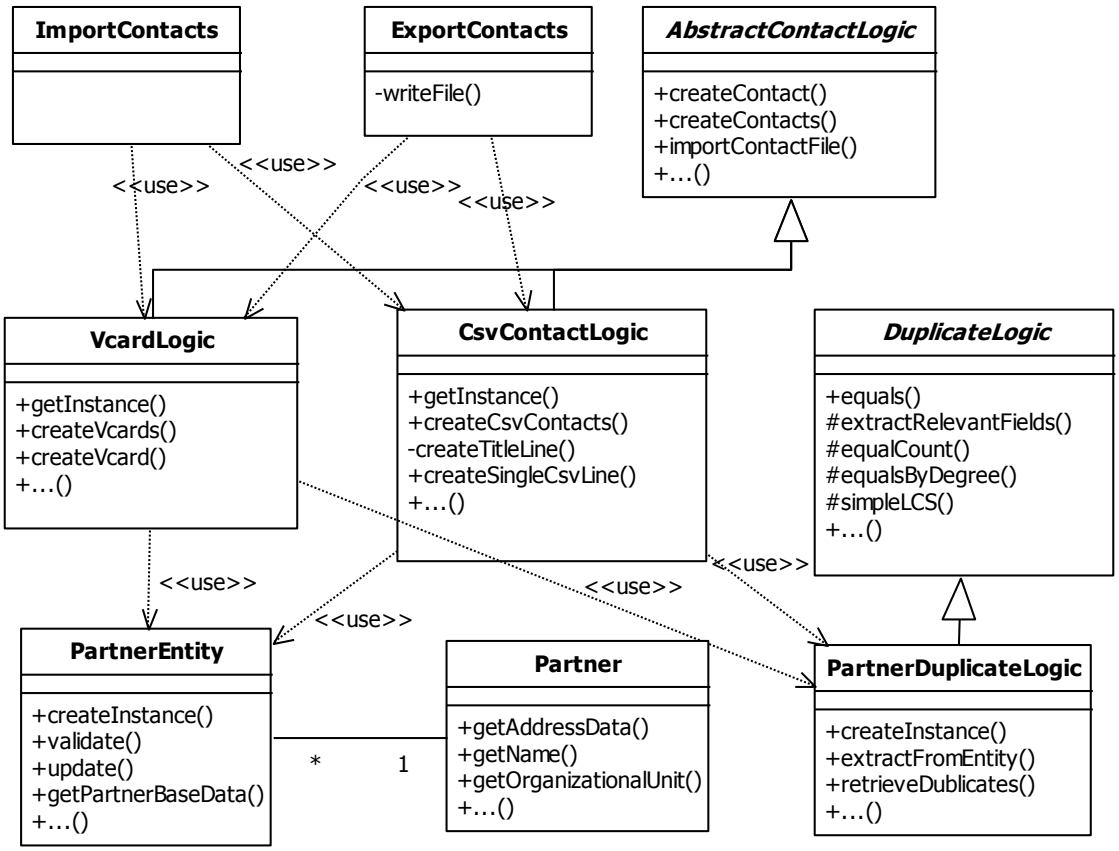


Figure 5.3.: UML class diagram of the Import/Export applications.

The SyncML application SyncWithSyncML is outlined in the class diagram in Figure 5.4. For the synchronization itself the SyncWithSyncML class uses the SyncMlManager. For passing all needed information to the SyncMlManager the class SyncMLConfig is used. It

contains all required information for the synchronization, like the SyncML server address or login data. The SyncMLConfig also has a list of SyncSource objects. A SyncSource is able to pass data to be synchronized of one specific type to the SyncMlManager. The SyncSource deals with the extraction, e.g., from database, and modification of data. One data item is represented by a SyncItem. A SyncItem object is uniquely defined by a key. In our case this key is the unique partner number, defined in Semiramis. It further has a state which indicates if the item is new, was updated or deleted. The SyncItem again can contain multiple SystemItemProperty objects for holding further data. The interface SyncItem is implemented by the class SyncItemImpl. The PartnerSyncSource implements the interface SyncSource. The PartnerSyncSource provides the partner data to be synchronized. For creating the SyncItem objects in the needed representation format, the PartnerSyncSource uses the VcardLogic class.

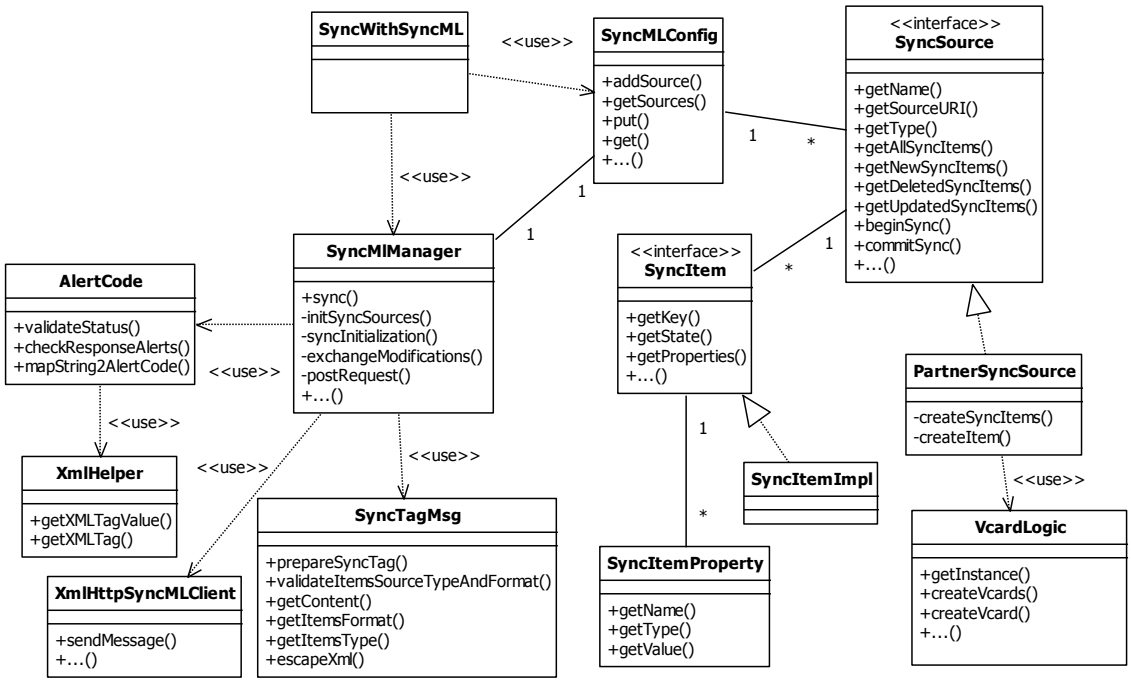


Figure 5.4.: UML class diagram of the SyncML background application.

The SyncMlManager has multiple helper classes which are: SyncTagMsg, XmlHttpSyncML-Client, and AlertCode. The SyncTagMsg class is used for the generation of SyncML XML

messages. On the other hand the class AlertCode is used for the interpretation of response SyncML XML messages from the SyncML server. It also validates if an error occurs. If that is the case an error message will be passed to Semiramis. The AlertCode class again uses the helper class XmlHelper. The class XmlHelper reads XML tags. The XmlHttpSyncMLClient is used by the SyncMlManager in order to send and receive XML messages to or from the server through HTTP or HTTPS.

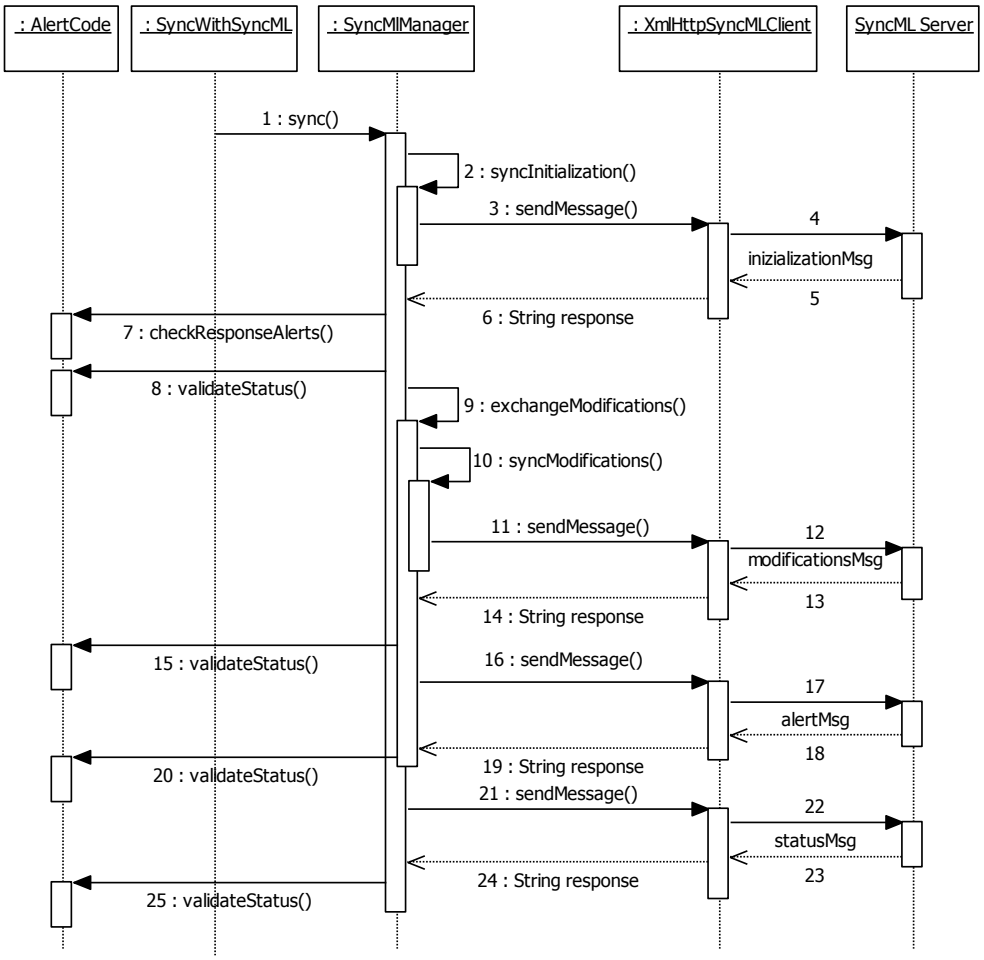


Figure 5.5.: UML sequence diagram of the SyncML synchronization process.

In Figure 5.5 the UML sequence diagram for the SyncML synchronization process can be seen. The process is started when the background application SyncWithSyncML invokes the sync method of the SyncMlManager. As previous mentioned the SyncMlManager uses the XmlHttpSyncMLClient to send messages to the SyncML server. The first message send is the initialization message, which contains the login data and the operations

to be performed. The server's answer contains information, if the authorization was granted and the operations are allowed. In the next message the SyncItems are sent to the server with one or more modification messages, depending on the configuration. The server notices if the items were newly added or were already stored. Then an alert message is sent to inform the server that all modifications were sent. In the next step the SyncMlManager could request and collect the modification from the server. But, since we are only doing a unidirectional synchronization this step is skipped. Finally, a status message is send to see if every thing went okay. Each time a respond of the SyncML server is received, the message is checked of the synchronization status. Thus, if an error occurs, the synchronization process could be halted.

6. Implementation

In this chapter the implementation of the prototype is described. First the prototype is presented with the help of some screenshots, then the implemented duplicate detection algorithm is explained. Finally, challenges which arose at the programming process are pointed out.

6.1. Prototype

Partner Cockpit Application. In Figure 6.1 the main partner cockpit application opened inside Semiramis can be seen. The red rectangle, the big rectangle at the top, highlights the selection fields of the application. Here the query for partners can be controlled. The green rectangle, the big rectangle at the bottom, points out the result list, here the query outcome is displayed. Beside each result item there is a checkbox, to enable the selection of specific result items for further processing. There is also a meta-checkbox to select all results. Since this cockpit application is a customizable cockpit, the user is able to choose selection and result fields freely. The complete set of selectable fields is depending on the OQL-search the cockpit is based on. Actually, the select and where clauses of the OQL-search are dynamical set depending on the customizing and filled selection fields. The main BO queried by the OQL-search is the BO Partner. In the OQL-search the BO Partner is joined with the BOs AddressData, Country, Region and others. Therefore, the fields of these BOs can also be queried. But, only if these fields are marked as displayable, they can be added to the GUI of the cockpit. The blue rectangle, the small rectangle at the top, shows the opened action list in the toolbar. In the action list the new actions, calling the background application for synchronizing, exporting, and importing contacts, can be found. When one of the background applications is invoked by clicking at the corresponding action, an adequate dialog is opened.

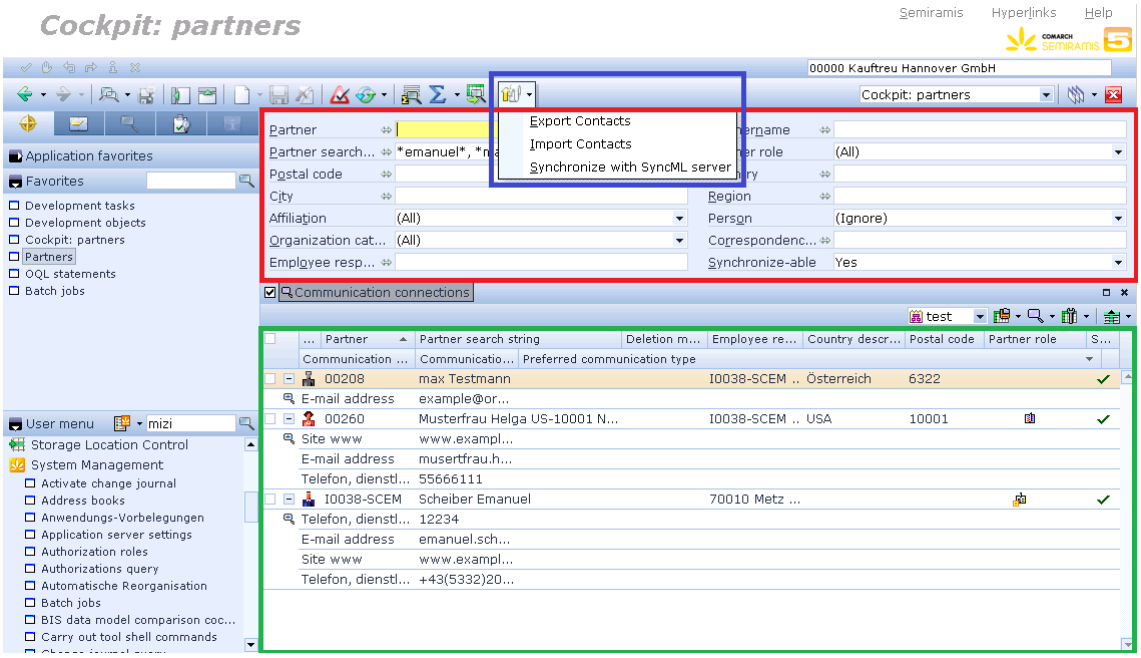


Figure 6.1.: Screenshot of the cockpit application.

Export Contacts Dialog. The dialog to export contacts can be seen in Figure 6.2. Here the user can first decide which selection should be considered. There are the options between *repeat database query*, *only the selected objects*, or *skip selected objects*. If, *only the selected objects* option is chosen, all partners selected through the check-boxes are exported. The option *skip selected objects* is the inverse selection of the previous mentioned one. The option *repeat database query* means that the check-boxes are ignored, instead the complete result of the query restricted by the selection fields will be exported. For information purpose only, the dialog shows which background application, defined by which development object, is invoked. Then, there is a parameter for the complete file path, including file name. In order to support the user, this field is implemented with a file system navigator. The file can be either located in the Semiramis knowledge store or on the local file system. The next parameter is *file format*, which can be selected by a drop-down list. The user can decide between vCard, or Outlook CSV file for the German or English version of Outlook. The *file encoding* parameter enables the user to chose between different encodings. However, usually the default setting does not have to be changed. There are three buttons to proceed further. The *cancel* button will cancel the action. The *immediately* button will instantly execute the action. The button *in batch* will execute the action according to the optional batch settings which

can be defined in the second tab of the dialog. These settings will be described later.

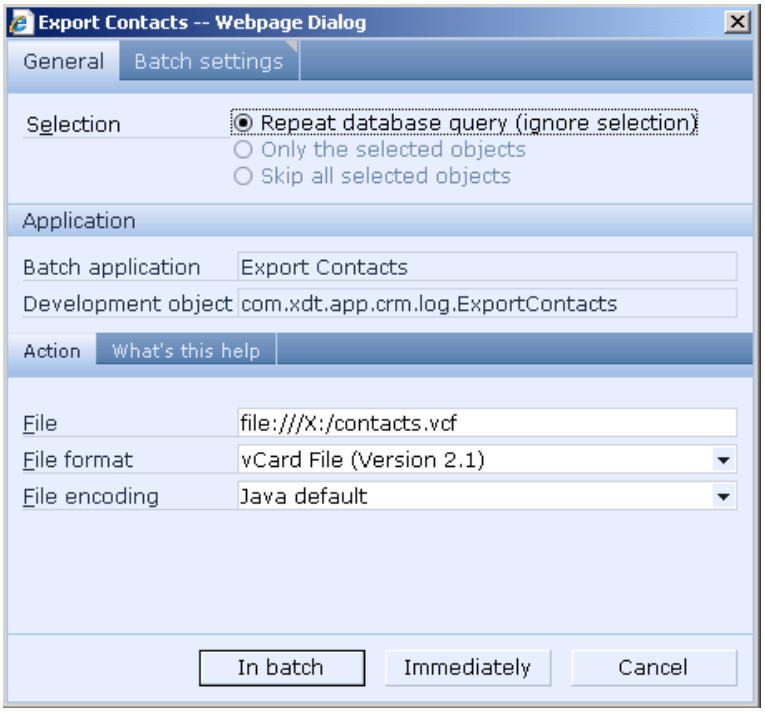


Figure 6.2.: Screenshot of the dialog for the background application export contacts.

Import Contacts Dialog. The dialog for the background application import contacts is shown in Figure 6.3. It is nearly equal to the previous described dialog. However, here the selection options have no influence at all. The only reason, why they are here is, because of the re-use of the generic batch applications dialog. Additionally, this dialog has the checkboxes *check for duplicates* and *confirm all warnings*. *Check for duplicates* indicates if the duplicate detection logic should be used or not. The checkbox *confirm all warnings* is relevant for the validation before the creation of new partner. When at the validation of partners warnings are thrown and the checkbox was not selected, then those partners will not be stored in Semiramis. Warnings are recommendations, but are not dangerous regarding the consistency or correctness of the data. For example, a warning is thrown if a telephone number is not in the recommended standardized format.

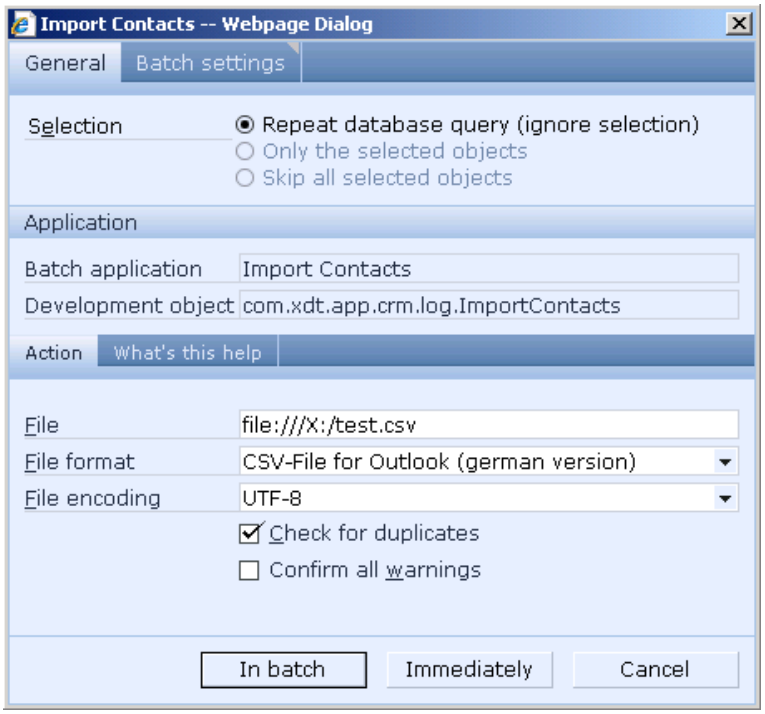


Figure 6.3.: Screenshot of the dialog for the background application import contacts.

Synchronize with SyncML Server Dialog. Figure 6.4 shows the dialog for the batch application synchronize with SyncML server. Here the selection options behave in the same way as in the export contacts dialog. The dialog includes some parameters needed for connecting to a SyncML server, which are: *SyncML server address*, *username*, *password*, and *contacts database*. This fields are prepopulated with the customizing entries which are later described. At default the checkbox *use the customizing settings* is ticked. When this checkbox is unselected, the user can overwrite the customizing settings.

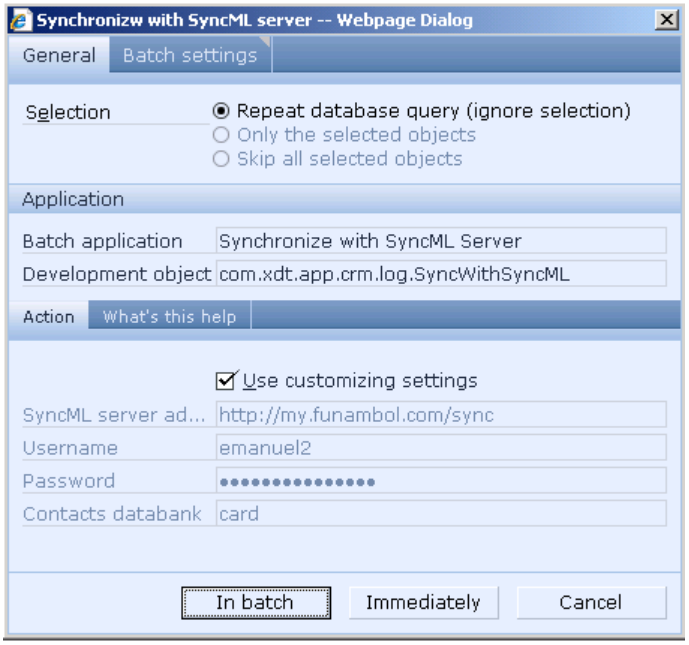


Figure 6.4.: Screenshot of the dialog for the background application synchronize with SyncML server.

Batch Settings Tab. In Figure 6.5 the background application tab *batch settings* can be seen on the left. It is the same for all three background applications. Here, the user can decide on which Semiramis application server and by which processing queue it should be executed. This setting especially plays a big role for resource expensive processes, since different application servers may have different workload and capacity. A processing queue sequently works off all assigned tasks. It can be selected when the batch application should be started, here, the options vary between *scheduled external*, *once only*, *as a series*, *at restart of application server*, or *immediately added to queue*. When the option *as a series* is chosen it can be specified in more detail by a serial pattern through the dialog which can be seen on the right of Figure 6.5. A further interesting parameter is *notification*. This is where the user can chose if he wants to be notified at the end of the execution of the background application and how. For example, notification can takes place by a dialog window which is shown only if errors arose or in any case.

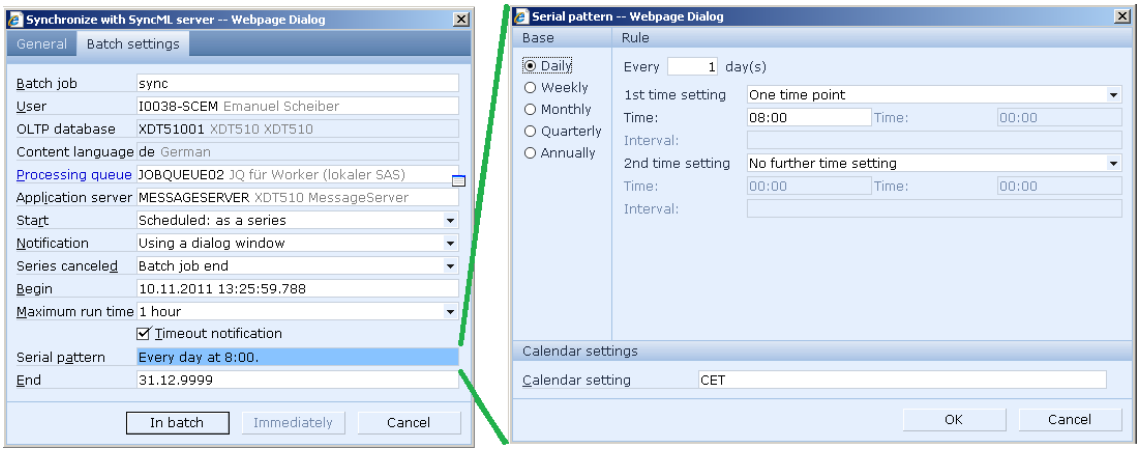


Figure 6.5.: Screenshots of the background applications dialogs batch settings.

Customizing Settings. In Figure 6.6 the customizing setting relevant for the partner synchronization application is shown. The import settings control the background application ImportContacts. The parameters *minimum number of identical fields* and *level of matching* are accessed by the PartnerDuplicateLogic. *Minimum number of identical fields* denotes how many fields of the two compared partners have to be equal to be considered as a duplicate. Whereas, the *level of matching* additionally states to which degree these fields have to be identical. The parameter *number range* is used for assigning unique partner numbers to the newly imported partners. The SyncML settings include the ones explained before at the dialog for the batch application synchronize with SyncML server, namely *server address*, *contact database* and all necessary login data. In addition, a *proxy server* and *proxy port* can be chosen. These may be needed for the HTTP connection with the SyncML server, if the address has to be accessed through a proxy.

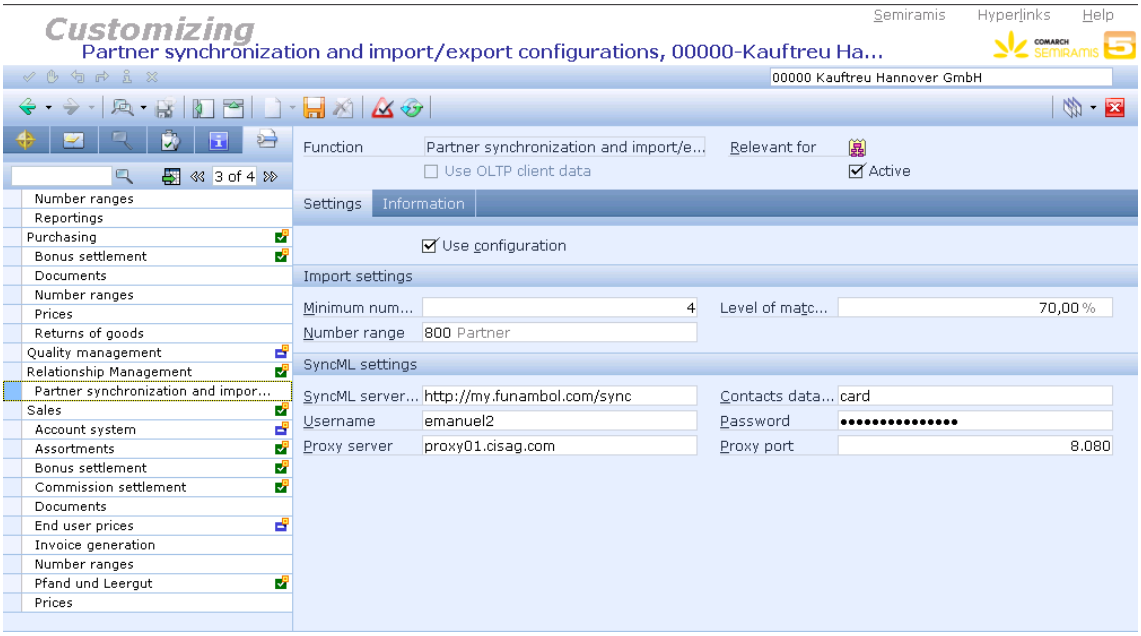


Figure 6.6.: Screenshots of the customizing settings.

6.1.1. Duplicate Detection Algorithm

In Subsection 2.4.2 we described how duplicate detection works in general. In this subsection we describe how this problem was solved for the import background application. Note, the complete process contains the steps: preprocessing of the data, formatting and partition of the data, determination of parameters, detection of duplicates, and fusion and integration of data.

(1) Preprocessing, Formatting and Partition of the Data. In our case the preprocessing of the data, formatting and partition is done by the PartnerEntity. The PartnerEntity and its contained Partner object is populated while reading the input file. The PartnerEntity only allows correctly formatted data to be set. Furthermore, the validation of the PartnerEntity verifies if the partner data is correct. Partition of the data does not take place.

(2) Determination of Parameters. The parameters chosen to be processed are: given name, middle name, surname, and the complete address data. In future it would be

possible that the user could be able to self-determine the parameters by adding a corresponding customizing setting.

(3) Detection of Duplicates. For the detection of duplicates the class PartnerDuplicateLogic is used, which was mentioned before in the design section. The main method in the PartnerDuplicateLogic for the duplicate detection is the method *retrieveDublicates()*. The process of the algorithm is following:

1. The partner object is converted into a hash table representation. That means all fields which are going to be compared against will stored in a hash table and thus, produce a partner representation which is quite easy and efficient to compare against others.
2. The method *retrieveDublicates()* is invoked with the created hash table.
3. A preliminary OQL search statement is created. This OQL search statement would, if unchanged, query all partners and returns those which have at least one attribute value in common.
4. From the statement an iterator or database cursor, respectively, is generated. This allows us to sequentially traverse the query result, in a resource saving manner.
5. A filter is attached to the iterator. This filter does the matching of the found records. The procedure for the matching is as following:

    a) In the first step all Guid fields are compared. Guid fields are fields which contain a reference to another instance of a BO. For example, the field *country* is such a field. Obviously, here it is not possible to take into account the degree ratio setting from the customizing.

    b) Then, all other field are compared against each other, now by taking into account also the degree ratio. To this end the simple longest common sequence algorithm is used, which only counts, for comparing two strings. It returns the number of characters that match in both compared strings.

    c) The number of matching characters is divided through the length of the longest of both compared strings in order to compute the degree of equality.

    d) If this degree is higher or equal then the one defined in the customizing, then the matching consider them as equal fields.

e) A partner is considered as match if the minimum number of identical fields defined in the customizing is reached.

6. With the filter attached the iterator only returns matching partners.
7. The method *retrieveDublicates()* only returns the iterator itself, so that it is possible to stop at the first found match or else to proceed and find all, or any other number of matches.

(4) Fusion and Integration of Data. The part fusion and integration of data is not implemented, since we discard a detected duplicate. This measure ensures us, that Semiramis is the leading system as claimed by the requirements.

6.2. Implementation Challenges and Restrictions

During the implementation several problems and challenges arose.

Different Formats. One big problem was that different PIMS create different vCard and CSV files. In the case of vCard the PIMSs export either in different versions, mostly vCard version 2.1 or 3, or have even an own version. For example, Outlook has an individual adaption of version 2.1 which defines some extra fields, like business telephone number. But, since there exists the vCard standard field for work telephone number, one could think the field business telephone number is redundant. Another special case originated from Google mails. Here the URL field was given different as expected by the vCard standard. When it comes to CSV files, there is a difference between the CSV files of the German version of Outlook and the English version, and maybe for other localization too. The overcome these issues the import logic is able to deal with the vCard version 2.1 and 3, and with English and German CSV files. Special vCard fields, which arose during the tests, are also processed correctly. However, there may exist still several cases which are not covered. But, the implementation is open to handle these cases, by simple adding additional if clauses for special cases. Thus, associating them with a corresponding partner attribute.

Business and Private Properties. A further problem is that Semiramis makes no difference between a business or private property in the background. One example is the work or private telephone number. In Semiramis the labels of communication data fields can be freely named. For instance, in some organization there may exist a communication data field labeled 'private telephone number' and another 'business telephone number'. In the database these are both stored as communication data of the communication method type telephone. This problem has been solved by retrieving the labels and if the substring 'privat' is found, the communication data field is guessed as a private field, else it is assumed to be a business field. This will only work for languages where this substring 'privat' makes sense, e.g., English and German. In the case of address data, the Semiramis standard label for the address data indicates that the partner address of a person is the private address and of an organization is the business address. It is not even possible to add a business address to a person. However, in any case it is exported as business address, since we simply assume that in the most case the private address is not given.

Parsing Attributes. Guessing whether an attribute is a business or private field is not the only guessing which has to be done. For example, at the import, in some cases the address is given by a single field instead of separated fields. In this case it is guessed that the address is given in the sequence: street, city, postal code, and country. A further issue is, that a country of an address data in Semiramis is an instance of the BO Country. That means, first the corresponding BO instance has to be found. At the moment this works only for the correct written country in either the language German or English. The problem is that an OQL search only queries the instances of a specific language. That means, for each relevant language the OQL search has to be done. However, it would be straight forward to implement it also for other languages, by adding further OQL search calls in the corresponding language.

SyncML Challenges. With the SyncML standard also some challenges come by. One issue is, that the SyncML XML messages are just very hard to interpret for the human being. Only so called alert codes, which are predefined numbers, indicate the status of the communication. For instance, the response messages saying that the authentication was successful or not are at the first glance quite identical, despite of the alert code. All in all, the SyncML synchronization is hard to debug, because errors returned by the SyncML server are just described by the alert code. So one can just vaguely say what

really happened at the server. A further problem of SyncML is that the servers databases name have to be known in advance and stated when requesting a synchronization. For instance, if contact data should be synchronized, the database name where contacts are stored at the server has to be known. SyncML describes no standard database names. Therefore, the database name for contacts have to be set, when calling the synchronization batch application.

7. Testing and Evaluation

In this chapter the implementation of the prototype and the conduction of the master thesis is evaluated. The prototype is evaluated by testing, while a design science research checklist will assess the approach of the master thesis.

7.1. Testing

Testing is used in order to evaluate the quality of the prototype implementation, by determine to which degree and how the specified requirements are satisfied. Use case based testing will be be applied for testing the functional requirements. In order to test non-functional requirements, we will observe each requirement individually.

7.1.1. Use Case Based Testing

In Section 4.5 we utilized use cases for the analysis and specification of functional requirements. Use cases provide a natural way of classifying the functional requirements of a software. Therefore, they lead directly to associated test suites of test cases in order to perform functional testing. For every use case we defined one main scenario and additional alternative scenarios. Each usage scenarios can act as an individual test case. Further scenarios can be created to add boundary value and error guessing test cases. In the following the test cases and their results are exposed. Additional facts like actors, preconditions, postconditions, covered requirements, which are not mentioned in the test cases, can be quoted from the related use case.

Table 7.1.: Test case 1: Correct synchronization of contact data.

Traceability Identifier	TC-1
Related Use Case	UC-2
Description	The main process of synchronizing contact data with error-free data and configuration.
Test Steps	**Begins** when user opens the application PartnerCockpit. 1. The user selects some partners. 2. The user starts the *synchronize with SyncML server* Semiramis background application. 3. Dialog parameters are correctly populated. 4. Button immediately is pressed. **Ends** when dialog closes and messages are displayed.
Expected Result	The partners are transmitted to the SyncML server and can now be synchronized with any PIMS.
Actual Result	As expected.
Pass/Fail	Pass

Table 7.2.: Test case 2: Synchronization of contact data fails due to faulty data.

Traceability Identifier	TC-2
Related Use Case	UC-2
Description	The main process of synchronizing contact data with faulty data.
Test Steps	**Begins** when user opens the application PartnerCockpit. 1. The user selects some partners which he is not authorized to synchronize. 2. The user starts the *synchronize with SyncML server* Semiramis background application. 3. Dialog parameters are correctly populated. 4. Button immediately is pressed. **Ends** when dialog closes and messages are displayed.
Expected Result	The process is aborted and no data is transmitted.
Actual Result	Unauthorized data is not shown to the user at the first place, thus cannot be transmitted.
Pass/Fail	Pass

Table 7.3.: Test case 3: Synchronization of contact data fails due to connection fail.

Traceability Identifier	TC-3
Related Use Case	UC-2
Description	The main process of synchronizing contact data with connection fail.
Test Steps	**Begins** when user opens the application PartnerCockpit. 1. The user selects some partners. 2. The user starts the *synchronize with SyncML server* Semiramis background application. 3. Dialog parameters are not correctly populated. 4. Button immediately is pressed. **Ends** when dialog closes and messages are displayed.
Expected Result	The process is aborted and no data is transmitted. A proper error message is displayed which hints what may be wrong.
Actual Result	As expected.
Pass/Fail	Pass

Table 7.4.: Test case 4: Correct export of partner data.

Traceability Identifier	TC-4
Related Use Case	UC-3
Description	The user correctly exports partner data, which may then be processed by any PIMS.
Test Steps	**Begins** when user opens the application PartnerCockpit. 1. The user selects some partners. 2. The user starts the *export contacts* Semiramis background application. 3. Dialog parameters are correctly populated. 4. Button immediately is pressed. **Ends** when dialog closes and messages are displayed.
Expected Result	A file in the chosen output format is generated and can be successfully processed by a PIMS.
Actual Result	As expected.
Pass/Fail	Pass

Table 7.5.: Test case 5: Export of partner data fails due to write permission deny.

Traceability Identifier	TC-5
Related Use Case	UC-3
Description	The user exports partner data without write permission.
Test Steps	**Begins** when user opens the application PartnerCockpit. 1. The user selects some partners. 2. The user starts the *export contacts* batch application. 3. Dialog parameters are correctly populated. 4. Button immediately is pressed. **Ends** when dialog closes and messages are displayed.
Expected Result	No file is generated and an error message is displayed.
Actual Result	As expected.
Pass/Fail	Pass

Table 7.6.: Test case 6: Import of correct partner data.

Traceability Identifier	TC-6
Related Use Case	UC-3
Description	The user imports partner data, originated from any PIMS.
Test Steps	**Begins** when user opens the application PartnerCockpit. 1. The user starts the *import contacts* Semiramis background application. 2. Dialog parameters are correctly populated. 3. Button immediately is pressed. **Ends** when dialog closes and messages are displayed.
Expected Result	New partners are successfully generated and persistent stored in Semiramis. Partners identified as duplicates, according to the settings, are not stored. An information message displays the assigned number of the new partner and points out which partners were detected as duplicates.
Actual Result	As expected.
Pass/Fail	Pass

Table 7.7.: Test case 7: Import of faulty partner data.

Traceability Identifier	TC-7
Related Use Case	UC-3
Description	The user imports incorrect partner data.
Test Steps	**Begins** when user opens the application PartnerCockpit. 1. The user starts the *import contacts* Semiramis background application. 2. Dialog parameters are correctly populated. 3. Button immediately is pressed. **Ends** when dialog closes and messages are displayed.
Expected Result	Partners with incorrect data are not stored in Semiramis, appropriate error or warning messages are thrown.
Actual Result	As expected.
Pass/Fail	Pass

Table 7.8.: Test case 8: Import of incorrect file.

Traceability Identifier	TC-8
Related Use Case	UC-3
Description	The user tries to import an incorrect file.
Test Steps	**Begins** when user opens the application PartnerCockpit. 1. The user starts the *import contacts* Semiramis background application. 2. Dialog parameters are correctly populated. 3. Button immediately is pressed. **Ends** when dialog closes and messages are displayed.
Expected Result	A warning message is thrown if the file extension is wrong, and thus prevents the execution at all. If the file extension itself is not wrong, no partners are stored and error messages are thrown.
Actual Result	As expected.
Pass/Fail	Pass

The above described test case cover all use cases, including the use cases' alternative scenarios. Since all test cases passed, it can be stated, that all functional requirements are satisfied.

7.1.2. Non-Functional Testing

Non-functional testing is the testing of a software application for its non-functional requirements. As mentioned before, we consider each requirement individually. In the following it is argued if and how each non-functional requirements is satisfied. The identifiers for the non-functional requirements correspond to those in Table 4.3.

- **NR1:** Since all exchange activities are triggered from a Semiramis application, it is made sure that Semiramis is the leading system.
- **NR2:** The Semiramis authorization functions, which are used by the prototype, ensure that only authorized data is accessed.
- **NR3:** The SyncML standard guarantees a neutral integration. Integration of any PIMS supporting SyncML is possible.
- **NR4, NR5, NR6, NR7:** It was sticked to style guides, guidelines, regulations, etc.
- **NR8, NR9, NR10:** Through SyncML client connectors and plugins, all different PIMS, surveyed during the requirement analysis, are covered. There exist such solutions for MS Outlook, iOS devices, Apple Mail, Hotmail, Yahoo!, Gmail, and much more PIMS.
- **NR11:** Semiramis serves as technical base for the prototype.
- **NR12:** Many mobile phones and their mobile PIMS support SyncML, including Android, BlackBerry, and iOS phones.
- **NR13:** The integration with a MS Exchange server is also possible with help of SyncML, for example the solution Synchronica can achieve that (Synchronica, 2011).
- **NR14:** The user manual is integrated and can be found in the standard online help tree of Semiramis.
- **NR15:** The standard Semiramis message queue is used for dealing with information, warning and error messages. Semiramis keeps protocol of all messages, thus allows to view those thrown in the background.

- **NR16:** The user can decide if it should be checked for duplicates or not. Furthermore, the customizing options allow the controlling of the detection algorithm.
- **NR17:** The customizing options allow the storage of SyncML server related information.
- **NR18:** HTTPS connection to a SyncML server is supported.
- **NR19:** The background settings dialog allows the time trigger invoking of the synchronization process.
- **NR20:** Additional training costs do not arise, because of its easy usage and the integrated user manual.
- **NR21:** It was developed with Semiramis version 5, it is only upwards compatible.
- **NR22:** Installation is easy because of the Semiramis integrated software delivery application. It is used for installing software updates or modifications.
- **NR23:** With the help of the background settings the synchronization process can safes a lot of resources.
- **NR24:** When a background application is started by the button in background, it runs in the background and thus does not interfere with the users other activities in Semiramis.
- **NR25:** The duplication detection's accuracy highly depends on the customizing settings. With strict settings duplicates can be prevented, but on the other side the number of wrong detected duplicates may increase. Depending on the settings, there will be always a trade off between them.
- **NR26:** The exported data stick to the vCard and CSV standards. Exported data were tested positive for successful processing in several PIMS.
- **NR27:** Imported data is beforehand validated by the standard validation, thus only correct data is stored in Semiramis.
- **NR28:** As mentioned before the accuracy of the duplicate detection depends on the settings.
- **NR29:** The prototype was successfully tested with different SyncML servers.
- **NR30:** The prototype's SyncML implementation is conform to the SyncML standard.
- **NR31:** The Partner attributes which are exported cover those from the requirements.

According to the above list, all non-functional requirements are fulfilled.

7.2. Design Science Research Checklist

In Section it was already mentioned that the design science research guidelines will be meet. In order to assess progress on design research projects and ensure that the projects address the key aspects of design science research Hevner and Chatterjee developed a checklist consisting of eight questions (Hevner and Chatterjee, 2010). In the following that checklist is used to evaluate the results of this thesis:

1. What is the research question (design requirements)? The software requirements were designed during the extensive requirements analysis in Chapter 4. The basic functional requirements were PIMS interfaces for the exchange of contacts, calendar information and emails with Semiramis. It was decided that the functionality to be implemented in the first prototype was the synchronization of contacts, because according to the customer survey and project meetings it had the highest priority. Additionally, 32 non-functional requirements were defined describing the softwares characteristics related to functionality, reliability, usability, efficiency, maintainability, and portability. Based on these requirements, use cases were constructed which describe how the functional requirements should be implemented. The use cases describe the operative scenarios of exporting, importing and synchronizing contact data, as well as the basic process of synchronizing PIM data. The core design problems incorporated in these requirements and use cases represent the essence of the design requirements and thus, the research question. The design problems were listed in Section 4.6. The main question was, if it is even possible to interchange PIM data between Semiramis and PIMS. Followed by the questions addressing the question of how and what.

2. What is the artifact? How is the artifact represented? On the one hand the main artifact is the implemented prototype. The prototype consists of over 40 Java classes and resulting in a total of 254 Semiramis development objects including help documents, error/warning/information messages, logical data types, data descriptions, object extensions, data views, string tables, actions, functions and text objects.

On the other hand there exist several further artifacts like the results of the requirements analysis, the comparison of similar approaches, the test and use cases, the UML diagrams, the user manual, the duplicate detection algorithm, etc., which were created during this master's project.

3. What design processes (search heuristics) will be used to build the artifact? The general design process used to build the prototype was the waterfall approach as described in Section 2.5.

According to the waterfall model, first the requirements were designed by a requirements analysis. Starting by drawing up the project constraints with the Comarch team, followed by pointing out external constraints by examining competitor products and the PIM market, and involving customers with an questionnaire, the projects baselines were worked out. After defining the system constraints, the functional and non-functional requirements were specified based on the previous input. In oder to determine operational scenarios for the prototype, use cases were constructed. At the end of the requirements analysis, the core design problems embedded in the requirements were summarized.

In the next phase the design and architecture of the prototype was conceptualized. Therefore, the topology of the application integration was outlined, which is determining the deployment. The prototypes architecture, application landscape and synchronization process was worked with the help of UML diagrams.

Following that, the coding and testing took place. Use case based testing was applied to verify the correct implementation of the functional requirements. Therefore, for each use case different test scenarios were deduced to ensure that all possible cases were covered. In order to test the non-functional requirements, each requirement individually was considered.

4. How are the artifact and the design processes grounded by the knowledge base? What, if any, theories support the artifact design and the design process? The theories and knowledge base on which the artifacts and the design processes are grounded on, are summarized in Chapter 2. The main domain are ERP systems and PIM, since this project extends the ERP Semiramis' PIM functionality. This is achieved by EAI, on a data integration level, therefore this is also a core topic. Since the quality of integrated data is also an important aspect of EAI, data quality, including its measurements like duplicate detection and conflict resolution, is also a core field of research representing the knowledge base. For the conduction of the project itself, theories about software development, including requirement elicitation, testing, etc., are also an essential part of the knowledge base. All in all, the design is based on well established standards and theories.

5. What evaluations are performed during the internal design cycles? What design improvements are identified during each design cycle? Based on a few loose ideas the requirements for the project were developed step by step. At first, the ideas were collected in a meeting with the Comarch team of Austria. Then, these were reviewed and extended by the product management of Comarch Germany. At this point, a large set of requirements for a PIM integration were collected. In order to learn the actual importance of them for customers, a survey was conducted. Only then, priorities were assigned to the requirements. Based on that, it was decided which functionality is implemented by the first prototype. Following that, first considerations on the design and realization of the prototype took place. Therefore, relevant technologies were examined and it was decided on the most promising ones to utilize. After fixating the basis technologies a concrete design was worked out. At first the prototype with the basis SyncML contact synchronization functionalities only was implemented detached from Semiramis. Meanwhile, the graphical part of the prototype with dummy functions was developed independently in Semiramis. The contact data import and export functionalities were added to this implementation. After finishing that, the SyncML component was merged with the Semiramis prototype, were only the data access methods had to be changed. While each sub-functionality was tested at its completion, the use case based testing took place at the end.

6. How is the artifact introduced into the application environment and how is it field tested? What metrics are used to demonstrate artifact utility and improvement over previous artifacts? For testing purposes the prototype was already introduced into an application environment. It is currently running a Semiramis temporary development system for interns, which served at the same time as implementation and test environment. However, in future the prototype will be established in other application environments. To ensure the correctness of the prototype in another environment, it makes sense to test it there again. Field testing should be conducted according to the test cases described in Section 7.1. So first, the SyncML synchronization process of contacts with correct and then faulty data should be tested. Test case 3, testing the behavior with connection fail, may be omitted, since the test result are independent from the application environment itself. Then the testing with export and import of both correct and faulty data should be conducted. The metrics used to demonstrate the artifact's utility are represented by the non-functional requirements formulated in Section 4.4.

7. What new knowledge is added to the knowledge base and in what form? To put it in a nutshell, the new knowledge added to the knowledge base is the content of the master thesis itself as well the prototype. The essence of the new knowledge can be stated as follows:

- The requirements to the integration.
- The PIM integration to ERP Semiramis with Semiramis as leading system.
- How the integration in Semiramis is performed.
- How the synchronization process with Semiramis and SyncML is performed.
- How a level of certain data quality is satisfied.
- How the duplicate prevention in Semiramis is performed.

The thesis summarized the requirements and their priority to the integration, which were developed by involving the Comarch team, customers and external factors. The project showed, how it is possible to integrate PIM to the ERP Semiramis, with emphasis on Semiramis taking the leading role. With the help of the SyncML standard, it was possible to synergize a wide variety of personal managers and Semiramis. Furthermore, a communication procedure was designed for broadcasting contacts from Semiramis to a SyncML server. For importing contacts, different measurements were adopted to maintain a high degree of data quality and prevent the generation of duplicates.

8. Has the research question been satisfactorily addressed? With the help testing it was ensured that the software requirements have been satisfactorily met. However, also the design problems, listed in Section 4.6, were covered while designing, implementing and evaluating the solution. The implemented prototype shows that it is possible to interchange PIM data between the ERP system Semiramis and various different PIMS. It can be achieved by data integration. We came to the conclusion that SyncML was best suited for that task. In order to make data interchangeable, a common data representation format had to been found. The most promising formats were vCard and CSV vor contact data, and XML for the communication protocol between systems. For programming the prototype, a lot of existing Semiramis functionalities could be reused. For example, for the graphical user interface the Semiramis standard cockpit application was utilized, and for the database access the Semiramis object manager was significant. The cockpit application template turned out to be a very practical way to select data which has to be exchanged. In order to prevent duplicate data entries an adequate algorithm with variable parameters was built in. All in all it can be said, that the requirements

as well as design problem have been satisfactorily addressed, since all design problems were tackled.

8. Conclusion and Future Work

The master thesis compromises the whole process of concreting an idea, designing requirements, implementing a solution and testing the prototype. In the following the results of the project is revised and and outlook for future work is given.

8.1. Conclusion

In this thesis the synergy between the ERP system Comarch Semiramis and personal information manager was examined. We noticed that at the present state it was not possible to exchange personal information, like contacts, emails, or appointments, between Semiramis and common personal information management systems. However, many customers of Comarch and other Semiramis users, pointed out that there was a need for a solution. So, it was decided to develop a first prototype for an enterprise application integration. As methodology for the master project's realization the design science research approach was complied, which is established for information systems projects. In the first step, literature and existing EAI approaches were reviewed in order to get an idea how to proceed with the project and furthermore how to design the prototype. The idea of integrating Semiramis and PIMS was at that point very abstract. Therefore, it was decided to conduct an extensive requirements analysis to consolidate well defined baselines. Stakeholders, like Comarch customers and employees, were included in the requirements design process by the help of meetings and a questionnaire. The outcome of the requirements analysis were 14 functional requirements, 32 non-functional requirements and four use cases. We decided that the first prototype should implement the exchange of Semiramis partner data, because stakeholders weighted this functionality with the highest priority. For that purpose we concluded that the data representation format, for importing and exporting contacts, to be supported is vCard and CSV. With these two data formats we can cover all surveyed PIMS. We decided that for the automatically synchronization of personal information data the standard SyncML is best

suited. SyncML was especially designed for synchronizing different personal information manager and is also well established for mobile devices. Thus, with the help of a SyncML server the prototype is now able to synchronize with nearly any PIMS. To ensure that Semiramis is the leading system, we implemented the synchronization only in one direction, from Semiramis to the SyncML server. After finishing the design and the implementation of the prototype, the testing took place. For that reason, test cases were deduced from the use cases. The results revealed that all functional requirements were met. Then, we checked the non-functional requirements and explained why those were also fulfilled. Finally, on the basis of the design science research checklist from Hevner and Chatterjee, the thesis results were interpreted.

During the master's project we learned that it is possible to integrate any kinds of systems, even though they were never meant to work together. It is because there exist information technologies, frameworks, and approaches for nearly every scenario. However, you have to ask yourself certain questions before actually developing an integration solution, like does it make sense, is value added, which information and data should be integrated, and how should be achieved, i.e., on which level. If we had to summarize the main problems of the master thesis in one word, the headword would be 'fragmentation'. Due, to fragmentation of applications and systems, information fragmentation arose. It is the single reason for the integration need. While developing the solution, we noticed that the fragmentation of standards makes it impossible to implement an 'one-size-fits-all' solution. The standards which are closed, proprietary, lacking user friendly documentations, are really hard to deal with. The questionnaire for surveying requirements pointed out the importance of mobile systems. Nowadays Internet capable mobile devices, like smart phones and tablet PCs, are wide spread. While planning a comprehensive solution for a software problem, this has to be kept in mind. During the project, we also learned the significance for both theory and praxis. With a well worked out concept and design, and with the technologies learned, most of the implementation is straight forward. Still, we stumbled across several problems in praxis, which could not be foreseen while thinking it theoretical through. For example, like difference in information depth or network issues like proxy servers. While working out the questionnaire, we learned that such a survey method makes only sense if you have narrowed down the problem and you know where you are heading. For implementing the prototype we would claim that object oriented programming languages, like Javas, are a very good way to go with. Because object oriented concepts, like subtype polymorphism or object inheritance, will save a lot of work while adapting the prototype with new functionality. In the next section we will go into more detail on what could be extended.

8.2. Future Work

The prototype implements only one basic functionality, so there is still a lot of future work to do. Possible extensions would be:

- The completion of calendar functionality, like integration with a PIMS calender and synchronization of appointments, tasks and activities.
- The addition of Email messages exchange, storage, attachment, and tagging capabilities.
- The implementation of a Semiramis integrated SyncML server. At the moment a third party SyncML server is needed for the synchronization. Semiramis resources, like databases and user authorization, could be used efficiently by integrating a SyncML server.

Bibliography

I. Alexander and N. Maiden. *Scenarios, stories, use cases: through the systems development life-cycle*. Wiley, 2004. ISBN 9780470861943.

R.V. Altekar. *Enterprisewide Resource Planning: Theory And Practice*. Prentice-Hall Of India Pvt. Ltd., 2006. ISBN 9788120326330.

Detlef Apel, Wolfgang Behme, Rüdiger Eberlein, and Christian Merighi. *Datenqualität erfolgreich steuern: Praxislösungen für Business-Intelligence-Projekte*. Hanser [u.a.], München, 2009. ISBN 9783446420564.

Deborah Barreau and Bonnie A. Nardi. Finding and reminding: file organization from the desktop. *SIGCHI Bull.*, 27:39–43, July 1995. ISSN 0736-6906.

M. Bell. *Service-oriented modeling: service analysis, design, and architecture*. John Wiley and Sons, 2008. ISBN 9780470141113.

Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

P. Bernus, L. Nemes, T.J. Williams, International Federation for Information Processing, and International Federation of Automatic Control. *Architectures for enterprise integration*. IFIP International Federation for Information Processing Series. Chapman and Hall, 1996. ISBN 9780412731402.

J.H. Blackstone, J.F. Cox, and APICS-The Educational Society for Resource Management. *APICS dictionary*. APICS, 2005. ISBN 9781558221956.

P. Bocij, A. Greasley, and S. Hickie. *Business information systems: technology, development and management*. Pearson Education, Limited, 2009. ISBN 9780273716624.

B. W. Boehm. A spiral model of software development and enhancement. *Computer*, 21 (5):61–72, May 1988. ISSN 00189162. doi: 10.1109/2.59.

B. Bond, Y. Genovese, D. Miklovic, B. Wood, B. Zrimsek, and N. Rayner. Erp is dead - long live erp ii. *Gartner Interactive*, 2000.

G. Booch, J. Rumbaugh, and I. Jacobson. *The unified modeling language user guide*. The Addison-Wesley object technology series. Addison-Wesley, 2005. ISBN 9780321267979.

campaignmonitor.com. Email client popularity: May 2011 @ONLINE, May 2011. URL `http://www.campaignmonitor.com/stats/email-clients/`. Retrieved October 24, 2011.

M. Castillo. *ERP II- Ein zukunftsfähiges Modell für Unternehmenssoftware?* GRIN Verlag GmbH, 2009. ISBN 9783640389674.

D. Chaffey and S. Wood. *Business Information Management: Improving Performance Using Information Systems*. Pearson Education, Limited, 2010. ISBN 9780273711797.

M. Cohn. *User stories applied: for agile software development*. The Addison-Wesley signature series. Addison-Wesley, 2004. ISBN 9780321205681.

Comarch. *Semiramis - Programmierhandbuch - Ab Semiramis 5*, 2010.

Comarch. Comarch semiramis - product overview - release 5.0. White paper, Strategy Partners International, 2011.

Versit Consortium. vcard the electronic business card version 2.1. Specification, sep 1996. URL `http://www.imc.org/pdi/vcard-21.txt`. Retrieved January 11, 2012.

Versit Consortium. vcard: The electronic business card. White paper, 1997. URL `http://www.imc.org/pdi/vcardwhite.html`. Retrieved January 11, 2012.

T.H. Davenport. *Process innovation: reengineering work through information technology*. Harvard Business School Press, 1993. ISBN 9780875843667.

F. Dawson and T. Howes. vcard mime directory profile. RFC 2426 (Proposed Standard), sep 1998. URL `http://www.ietf.org/rfc/rfc2426.txt`. Retrieved January 11, 2012.

P.F. Drucker. *Management: tasks, responsibilities, practices*. Drucker series. Butterworth-Heinemann, 1999. ISBN 9780750643894.

Halbert L. Dunn. Record linkage. *American Journal of Public Health*, 12(36):1412–1416, 1946.

D. Ehrenberg and M. Ewig. Collaboratives business - eine herausforderung für die wirtschaftsinformatik. In K. Fink and C. Ploder, editors, *Wirtschaftsinformatik als Schlüssel zum Unternehmenserfolg*. Deutscher Universitätsvlg, 2006. ISBN 9783835002937.

Ergin Elmacioglu, Min-Yen Kan, Dongwon Lee, and Yi Zhang. Web based linkage. In *Proceedings of the 9th annual ACM international workshop on Web information and data management*, WIDM '07, pages 121–128, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-829-9.

Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. Duplicate record detection: A survey. *IEEE Trans. on Knowl. and Data Eng.*, 19:1–16, January 2007. ISSN 1041-4347.

H. Engesser, V. Claus, and A. Schwill. *Duden Informatik: ein Sachlexikon für Studium und Praxis*. Duden Series. Dudenverlag, 1993. ISBN 9783411052325.

C. Fellbaum. *WordNet: an electronic lexical database*. Language, speech, and communication. MIT Press, 1998. ISBN 9780262061971.

I. P. Fellegi and A. B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64:1183–1210, 1969.

L. L Floridi. Is information meaningful data? *PHILOSOPHY AND PHENOMENOLOGICAL RESEARCH*, 70:351—370, 2000.

Helmuth Gümbel. Semiramis - the native next generation business software. white paper - version 5.0. White paper, Strategy Partners International, 2005.

N. Gronau. *Enterprise Resource Planning und Supply Chain Management*. Lehrbücher Wirtschaftsinformatik. Oldenbourg, 2004. ISBN 9783486272659.

Janis Grundspenkis. Agent based approach for organization and personal knowledge modelling: knowledge management perspective. *Journal of Intelligent Manufacturing*, 18:451–457, 2007. ISSN 0956-5515.

Anna Lisa Guido and Roberto Paiano. Semantic integration of information systems. *International journal of Computer Networks and Communications*, 2:48–64, 2010. ISSN 09752293.

Alon Y. Halevy, Naveen Ashish, Dina Bitton, Michael Carey, Denise Draper, Jeff Pollock, Arnon Rosenthal, and Vishal Sikka. Enterprise information integration: successes, challenges and controversies. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, SIGMOD '05, pages 778–787, New York, NY, USA, 2005. ACM. ISBN 1-59593-060-4.

Richard W. Hamming. Error detecting and error correcting codes. *Bell Syst. Tech. J*, 29(2):147–160, 1950.

Marti A. Hearst. Information integration. *IEEE Intelligent Systems*, 13:12–24, 1998. ISSN 1094-7167.

B. Heinrich, M. Klier, and M. Kaiser. A procedure to develop metrics for currency and its application in crm. *J. Data and Information Quality*, 1:5:1–5:28, June 2009. ISSN 1936-1955.

T.N. Herzog, F. Scheuren, and W.E. Winkler. *Data quality and record linkage techniques*. Springer eBooks collection: Computer science. Springer, 2007. ISBN 9780387695020.

Alan Hevner and Samir Chatterjee. *Design Research in Information Systems*. Springer, New York, NY, USA, 2010. ISBN 978-1-4419-5652-1.

S. March J. Park Hevner, A. and S. Ram. Design science in information systems research. *MIS Quarterly*, 28:75–105, 2004.

IEEE Computer Society. *Software Engineering Body of Knowledge (SWEBOK)*. Angela Burgess, EUA, 2004. URL `http://www.swebok.org/`.

ISO/IEC. *ISO/IEC 9126. Software engineering – Product quality*. ISO/IEC, 2001.

F. Robert Jacobs, F.C. 'Ted' Weston, and Jr. Enterprise resource planning (erp)–a brief history. *Journal of Operations Management*, 25(2):357 – 363, 2007. ISSN 0272-6963. Special Issue Evolution of the Field of Operations Management SI/ Special Issue Organisation Theory and Supply Chain Management.

A. Jetter and J. Kraaijenbrink. *Knowledge integration: the practice of knowledge management in small and medium enterprises*. Physica-Verlag, 2006. ISBN 9783790815863.

William Jones. Personal information management. *Annual Review of Information Science and Technology*, 41(1):453–504, 2007. ISSN 1550-8382.

W.P. Jones and J. Teevan. *Personal information management*. University of Washington Press, 2007. ISBN 9780295987378.

R. Jung. *Architekturen zur Datenintegration: Gestaltungsempfehlungen auf der Basis fachkonzeptueller Anforderungen*. Deutscher Universitätsvlg, 2006. ISBN 9783835002432.

M. Kaib. *Enterprise Application Integration: Grundlagen, Integrationsprodukte, Anwendungsbeispiele*. Deutscher Universitäts-Verlag, 2002. ISBN 9783824421633.

Brett Kessler. Phonetic comparison algorithms. *Transactions of the Philological Society*, 103(2):243–260, 2005. ISSN 1467-968X.

H. Krcmar. *Informationsmanagement*. Springer, 2003.

K. Kurbel. *Produktionsplanung und-steuerung im Enterprise Resource Planning und Supply Chain Management*. Lehrbücher Wirtschaftsinformatik. Oldenbourg, 2005. ISBN 9783486575781.

M. Lansdale. The psychology of personal information management. *Applied Ergonomics*, 19(1):55–66, March 1988. ISSN 00036870.

P.A. Laplante. *Requirements Engineering for Software and Systems*. Auerbach Series on Applied Software Engineering. CRC Press, 2009. ISBN 9781420064674.

YoungSeok Lee, YounSoo Kim, and Hoon Choi. Conflict resolution of data synchronization in mobile environment. In *Computational science and its applications: ICCSA 2004*, Lecture notes in computer science, pages 196–205. Springer, 2004. ISBN 9783540220565.

Franz Lehner and Ronald Maier. Can information modelling be successful without a common perception of the term information? pages 181–198. 1998. ISBN 90-5199-334-X.

Maurizio Lenzerini. Data integration: a theoretical perspective. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '02, pages 233–246. ACM, 2002. ISBN 1-58113-507-6.

J. Leonard and Defense Systems Management College. Press. *Systems engineering fundamentals: supplementary text*. Defense Systems Management College, 1999. ISBN 9781428980976.

Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. Technical Report 8, 1966.

Bennett P. Lientz and E. Burton Swanson. *Software Maintenance Management.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1980. ISBN 0201042053.

D.S. Linthicum. *Enterprise application integration.* Addison-Wesley information technology series. Addison-Wesley, 2000. ISBN 9780201615838.

litmus.com. Email client market share @ONLINE, 2010. URL `http://litmus.com/resources/email-client-stats`. Retrieved October 23, 2011.

Ravindranath Madhavan and Rajiv Grover. From embedded knowledge to embodied knowledge: New product development as knowledge management. *The Journal of Marketing*, 62:1–12, 1998.

R. Maier. *Knowledge Management Systems: Information and Communication Technologies for Knowledge Management.* Springer, 2007. ISBN 9783540714071.

R. Maier, T. Hädrich, and R. Peinl. *Enterprise Knowledge Infrastructures.* Springer, 2009. ISBN 9783540897675.

D. Marca and G. Bock. *Groupware: software for computer-supported cooperative work.* IEEE Computer Society Press tutorial. IEEE Computer Society Press, 1992. ISBN 9780818626371.

S. March and G. Smith. Design and natural science research on information technology. *Decision Support Systems*, 15:251–266, 1995.

P. Mertens and G. Knolmayer. *Organisation der Informationsverarbeitung.: Grundlagen - Aufbau - Arbeitsteilung.* Gabler, 1998. ISBN 9783409331067.

Michael Robert Middleton. *A conceptual framework for information management: formation of a discipline.* Dissertation, Queensland University of Technology, 2006.

Kyongho Min and William H. Wilson. Syntactic recovery and spelling correction of ill-formed sentences. In *3rd Conference of the Australasian Cognitive Science (CogSci95*, page 82. Ablex, 1995.

Alvaro Monge and Charles Elkan. An Efficient Domain-Independent Algorithm for Detecting Approximately Duplicate Database Records. In *SIGMOD workshop on data mining and knowledge discovery*, May 1997.

Kazuo Nakatani, Ta-Tao Chuang, and Duanning Zhou. Data synchronization technology: Standards, business values and implications. *Communications of the Association for Information Systems*, 17(1):2–60, 2006.

R. Nisbet, J. Elder, J.F. Elder, and G. Miner. *Handbook of statistical analysis and data mining applications*. Academic Press. Academic Press/Elsevier, 2009. ISBN 9780123747655.

Mark J. O'Brien and Ali Al-Soufi. Electronic data interchange and the structure of the uk construction industry. *Construction Management and Economics*, 11(6):443, 1993. ISSN 01446193.

D.L. Olson and D. Delen. *Advanced data mining techniques*. Springer, 2008. ISBN 9783540769163.

OMA. Oma syncml common specification v1.2.2, July 2009. URL `http://www.openmobilealliance.org/Technical/release_program/SyncML_v1_2_2.aspx`. Retrieved January 11, 2012.

Tim O'Reilly. What is web 2.0: Design patterns and business models for the next generation of software. MPRA Paper 4578, University Library of Munich, Germany, March 2007.

D.A. Patterson and J.L. Hennessy. *Computer organization and design: the hardware/-software interface*. The Morgan Kaufmann Series in Computer Architecture and Design. Elsevier Morgan Kaufmann, 2008. ISBN 9780123744937.

T.J. Peters and R.H. Waterman. *In search of excellence: lessons from America's best-run companies*. Collins Business Essentials. Springer, 2004. ISBN 9780060548780.

Leo L. Pipino, Yang W. Lee, and Richard Y. Wang. Data quality assessment. *Commun. ACM*, 45:211–218, April 2002. ISSN 0001-0782.

M E Porter and V E Millar. How information gives you competitive advantage. *Harvard Business Review*, 63(4):149–160, 1985.

Hans Joachim Postel. Die kölner phonetik. ein verfahren zur identifizierung von personennamen auf der grundlage der gestaltanalyse. *Transactions of the Philological Society*, 103(2):243–260, 2005. ISSN 1467-968X.

A. Prencipe, A. Davies, M. Hobday, and M. Hobday. *The business of systems integration*. Oxford scholarship online. Oxford University Press, 2005. ISBN 9780199263233.

T.C. Redman. *Data quality for the information age.* Artech House Computer Science Library. Artech House, 1996. ISBN 9780890068830.

Thomas C. Redman. The impact of poor data quality on the typical enterprise. *Communications of The ACM*, 41:79–82, 1998. doi: 10.1145/269012.269025.

W. W. Royce. Managing the development of large software systems: concepts and techniques. In *Proceedings of the 9th international conference on Software Engineering*, ICSE '87, pages 328–338, Los Alamitos, CA, USA, 1987. IEEE Computer Society Press. ISBN 0-89791-216-0.

W.A. Ruh, F.X. Maginnis, and W.J. Brown. *Enterprise application integration: a Wiley tech brief.* Wiley tech brief series. John Wiley, 2001. ISBN 9780471376415.

Bernd Scholz-Reiter and Jens Jakobza. Supply chain management - Überblick und konzeption. *HMD - Praxis der Wirtschaftsinformik*, 207:7–15, 1999.

Frank Schultz. Evaluation von open source projekten: Ein gqm-basierter ansatz. Diplomarbeit, Technische Universität München, May 2003.

semiramis.com. Semirams@ONLINE, June 2011. URL `http://www.semiramis.com/`.

Y. Shafranovich. Common format and mime type for comma-seperated values (csv) files. RFC 4180 (Proposed Standard), October 2005. URL `http://www.ietf.org/rfc/rfc4180.txt`. Retrieved January 11, 2012.

H. Stadtler and C. Kilger. *Supply chain management and advanced planning: concepts, models, software and case studies.* Springer, 2005. ISBN 9783540220657.

Street N W Suite. Software as a service: Strategic backgrounder. *Online*, 21(February): 1–18, 2001.

Synchronica. Synchronica website@ONLINE, June 2011. URL `http://www.synchronica.com`. Retrieved January 11, 2012.

Giri Kumar Tayi and Donald P. Ballou. Examining data quality. *Commun. ACM*, 41: 54–57, February 1998. ISSN 0001-0782.

The National Archives. The soundex indexing system, May 2007. URL `http://www.archives.gov/research/census/soundex.html`. Retrieved January 11, 2012.

V. Torra. *Information fusion in data mining.* Studies in fuzziness and soft computing. Springer, 2003. ISBN 9783540006763.

T.E. Vollmann. *Manufacturing planning and control for supply chain management.* Irwin/McGraw-Hill series in operations and decision sciences. McGraw-Hill/Irwin, 2005. ISBN 9780071121330.

J. G. Walls and G. R. Widmeyer. Building an information system design theory for vigilant eis. *Information Systems Research*, 3:36–59, 1992.

Richard Y. Wang and Diane M. Strong. Beyond accuracy: what data quality means to data consumers. *J. Manage. Inf. Syst.*, 12:5–33, March 1996. ISSN 0742-1222.

R.Y. Wang, Y.Y.R. Wang, E.M. Pierce, S.E. Madnick, and C.W. Fisher. *Information quality.* Advances in management information systems. M. E. Sharpe, 2005. ISBN 9780765611338.

F. Wijnhoven. *Information Management: An Informing Approach.* Taylor and Francis, 2009. ISBN 9780415552141.

D. Wood. *Linking Enterprise Data.* Springer, 2010. ISBN 9781441976642.

Ping Wu and Stanley Y. W. Su. Rule validation based on logical deduction. In *Proceedings of the second international conference on Information and knowledge management*, CIKM '93, pages 164–173, New York, NY, USA, 1993. ACM. ISBN 0-89791-626-3.

L. Wylie. A vision of next generation mrp ii. Scenario S-300-339, April 1990.

Su Yan, Dongwon Lee, Min-Yen Kan, and Lee C. Giles. Adaptive sorted neighborhood methods for efficient record linkage. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '07, pages 185–194, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-644-8.

A. Questionnaire

The questionnaire was distributed among several Semiramis users, customers as well as Semiramis professionals. There were 13 participants. The area of work of the participants encompasses system administration, consulting, IT management, controlling, partner management and product management. On average the participants work about 6 hours per working day with Semiramis. Only one participants has indicated that his organization has started a project of synchronizing Semiramis with a PIMS.

The questionnaire is structured into general, PIMS, synchronization application, and data quality questions. The general questions is about getting to know how the participant is related to Semiramis. The next part is to learn about the participant's personal information manager habits. Following that, the participant can express the importance of possible functionalities for a Semiramis synchronization application. Finally, the participant can rate the significance of the data quality dimension regarded to the application. In the following the questionnaire is shown.

Questionnaire for Integration of Personal Information Manager and Semiramis

---

In which organization are you employed?

What is your position or function in the organization?

The average time in hours you are using Semiramis during a workday ?

Personal Information Management

Which personal information manager / E-Mail-clients are you using at work?
Multiple answers are possible.

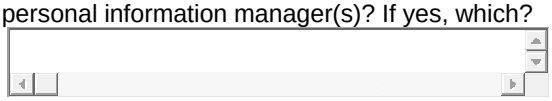
- ☐ Microsoft Outlook (Version 2003 or newer)
- ☐ Microsoft Outlook (Version 2002 or older)
- ☐ Mozilla Thunderbird
- ☐ Lotus Organizer
- ☐ Lotus Notes
- ☐ Google Mail
- ☐ Apple iCal
- ☐ Microsoft Outlook Web App / Outlook Web Access
- ☐ Microsoft Outlook Express
- ☐ Other:

Which mobile personal information manager are you using at work?

Which functions of your personal information manager / E-Mail-clients are you using at work?
Multiple answers are possible.

- ☐ Sending and receiving of E-Mails
- ☐ Contact management (maintain contacts and addresses)
- ☐ Time management (manage appointments, tasks, activities, etc.)
- ☐ Document and memo management
- ☐ Synchronization and data exchange with mobile devices
- ☐ Synchronization and data exchange with other systems
- ☐ Project management (milestones, progress, etc.)
- ☐ Subscription and reading of news (RSS feeds, newsgroups, etc.)
- ☐ Customer-Relationship-Management
- ☐ Knowledge management (wiki, knowledge database, etc.)
- ☐ Computer aided group work (use as groupware, collaborative writing, etc.)
- ☐ Other:

Do you use technologies to exchange data between Semiramis and your personal information manager(s)? If yes, which?



Semiramis Synchronization Application

How important are following functions for a possible Semiramis synchronizations application for you?
1 = not important, 4= very important

	1	2	3	4
Contacts/ Partner synchronization	○	○	○	○
Calendar / Appointments/ Tasks / Activities synchronization	○	○	○	○
Storage of E-Mails in Semiramis	○	○	○	○

	1	2	3	4
Duplicate detection / Record linkage	○	○	○	○
Options for conflict resolution	○	○	○	○
Manual invoke of the data synchronization	○	○	○	○
Scheduled invoke of the data synchronization	○	○	○	○
Tagging of imported E-Mails	○	○	○	○
Jumping from a Semiramis date field into the calendar of a personal information manager	○	○	○	○
Jumping from a Semiramis workflow activity into the calendar of a personal information manager	○	○	○	○
Neutral integration without fixation on a specific personal information manager	○	○	○	○
Bidirectional synchronization of data	○	○	○	○
Unidirectional synchronization of data from Semiramis to personal information manager	○	○	○	○
Unidirectional synchronization of data form a personal information manager to Semiramis	○	○	○	○
Attachment of E-mails in Semiramis as documents to several business objects	○	○	○	○
Document exchange	○	○	○	○

Which other function are desirable for a possible Semiramis synchronization application?

Which technical solution would you prefer?

- Semiramis - Microsoft Exchange Server (synchronization on server level, with a synchronization application running on Semiramis)
- Semiramis - Personal Information Manager (synchronization on client level, with a synchronization application running on Semiramis)
- Middleware (synchronization on client level, with a other standalone application as synchronization application)
- Semiramis - Personal Information Manager Plug-in (synchronization on client level, with for example an Outlook add-on)
- Manual synchronization (relevant data is imported and exported by the user from the corresponding systems)
- Other:

Data Quality

How important are the following data quality dimensions in relation of a possible Semiramis synchronization application in your opinion?
1 = not important, 4= very important

	1	2	3	4
Accessibility				
Appropriate amount of data				
Believability				
Completeness				
Concise representation				
Consistent representation				
Ease of manipulation				
Free of error				

	1	2	3	4
Interpretability	○	○	○	○
Objectivity	○	○	○	○
Relevancy	○	○	○	○
Reputation	○	○	○	○
Timeliness	○	○	○	○
Understandability	○	○	○	○
Value-added	○	○	○	○

B. User Manual

At the moment the user manual is only available in German. In future it may be translated in other languages, depending on the customers. The user manual consist of a general documentation of the partner cockpit application and a procedure documentation showing one specific use case of synchronizing data with Microsoft Outlook via SyncML.

Betriebswirtschaftliche Dokumentation

Erweiterung: Cockpit: Partner und Partner Synchronisation

Release	Ab Semiramis 5
Ausgabedatum	03/2012
Referenz auf andere Dokumente	Partner Cockpit: Partner Bedienungsleitfaden Berechtigungen

Inhaltsverzeichnis

1 Themenübersicht

Die Anwendung „Cockpit: Partner" dient zur Suche, Anzeige und Auswahl von Partnern. In dieser Anwendung können Sie nach allen im System erfassten Partnern suchen, unabhängig von deren Partnertyp oder Partnerrolle. Sie können Partner als vCard oder CSV Datei exportieren, importieren und Partner mit einem SyncML Server synchronisieren.

2 Begriffsbestimmung

Partner

Ein Partner ist eine natürliche Person oder eine Organisation. Ein Partner kann sowohl mit dem eigenen Unternehmen (= Mandant, der der aktuell verwendeten OLTP-Datenbank zugeordnet wurde) als auch mit anderen Partnern in einer Beziehung (Partnerbeziehung) stehen. Dadurch kann der Partner in verschiedenen Partnerrollen vorkommen und in unterschiedliche Geschäftsprozesse eingebunden sein (Partnerverwendungen).

Kontakte

Kontakte sind Partner, ohne Kunden-, Lieferanten- oder Mitarbeiterinformationen. Enthalten sind Informationen über Name, Titel, Adresse und Kommunikationsverbindungen.

3 Anwendungsbeschreibung

Mit der Anwendung „Cockpit: Partner" verschaffen Sie sich einen Überblick über die erfassten Partner, um sie im Rahmen des Beziehungs-Managements weiterzubearbeiten. Die Anwendung hat dazu einen Abfrage- und einen Arbeitsbereich.

3.1 Abfragebereich

Im Abfragebereich werden Abfragefelder abgelegt. In der Regel sind die wichtigsten Abfragefelder bereits vorhanden. Die Anzahl der Abfragefelder und deren Platzierung kann jedoch variieren.

Der Abfragebereich ist für alle Ansichten des Arbeitsbereichs gleich. Welche Abfragefelder angezeigt werden, können Sie selbst bestimmen.

Nachfolgend sind alle wesentlichen Abfragefelder aufgeführt.

Feld	**Erläuterung**
Marketingpartner	Geben Sie hier bei Bedarf die Partnernummer zur Identifikation der gesuchten Partner an.
Name	Erfassen Sie hier bei Bedarf einen Namen als Suchmerkmal.
Partnersuchbegriff	Erfassen Sie hier bei Bedarf einen Suchbegriff als Suchmerkmal.
Partnerrolle	Wählen Sie hier bei Bedarf eine oder mehrere Partnerrollen als Suchmerkmal für den gesuchten Partner aus. Sie können eine, mehrere oder alle der folgenden Partnerrollen und Partnerverwendungen auswählen. Partnerrollen: • „Kontakt" • „Kunde" • „Lieferant" • „Mitarbeiter" • „Mitbewerber" • „Vertreter" • „Ansprechpartner" • „Spediteur"
PLZ	Erfassen Sie eine Postleitzahl als Suchmerkmal für die Partner.
Land	Wählen Sie ein Land als Suchmerkmal für die Partner.
Ort	Erfassen Sie einen Ort als Suchmerkmal für die Partner.
Region	Wählen Sie eine Region als Suchmerkmal für die Partner.
Zugehörigkeit	Wählen Sie bei Bedarf, ob der gesuchte Partner ein interner oder ein externer Partner ist.
Person	In diesem Feld können Sie bei Bedarf die Suche auf Partnertyp „Person" (Ja) bzw. „Organisation" (Nein) einschränken. Als Vorschlagswert wird der Partnertyp ignoriert.
Organisationstyp	Wählen Sie bei Bedarf eine Organisationsform als Suchmerkmal aus. Nur Partner des Typs Organisationen verwenden eine Organisationsform. Sie können einen, mehrere oder alle der folgenden Organisationsformen wählen: • „Unternehmen" • „Geschäftsbereich" • „Niederlassung" • „Abteilung" • „Gruppe" • „Pseudopartner"

Feld	Erläuterung
Korrespondenzsprache	Wählen Sie bei Bedarf eine im System vorhandene Sprache als Suchmerkmal für den Partner.
Zuständiger Mitarbeiter	Erfassen Sie bei Bedarf als Suchmerkmal den zuständigen Mitarbeiter. Zulässig sind externe und interne Partner des Typs Person und der Rolle Mitarbeiter.
Beziehungstyp	Erfassen Sie einen Beziehungstyp als Suchmerkmal für die Partner.
Landbezeichnung	Geben Sie hier bei Bedarf die Landbezeichnung zur Identifikation der gesuchten Partner an.
Löschkennzeichen	Geben Sie bei Bedarf als Suchmerkmal an, ob der Marketingpartner mit einem Löschkennzeichen versehen wurde. Sie können wählen zwischen: • „Mit Löschkennzeichen" • „Ohne Löschkennzeichen" • „Alle" (Vorschlagswert)

3.2 Arbeitsbereich

Im Arbeitsbereich steht eine Tabelle zur Verfügung, in der die abgefragten Partner angezeigt werden. Welche Informationen zu einem Partner angezeigt werden, können Sie selbst bestimmen. Um einen Partner auf verschiedene Weise mit unterschiedlichen Informationen betrachten zu können, können mehrere Ansichten erstellt und abgespeichert werden.

3.2.1 Ansicht „Standard"

Nachfolgend sind alle wesentlichen Tabellenspalten dieser Ansicht beschrieben.

Spalte	Erläuterung
Typ	Unter „Typ" wird ein Icon angezeigt. Das Icon verdeutlicht den Partnertyp (Person oder Organisation) sowie die Zugehörigkeit (intern/extern). Bei Organisationen wird außerdem der Organisationstyp ersichtlich (Unternehmen, Geschäftsbereich, Standort, Abteilung, Gruppe oder Pseudopartner). Siehe auch Liste der verwendeten Symbole und Icons im Bedienungsleitfaden.
Partner	Die Partnernummer und der Suchbegriff laut Partner-Basisdaten werden angezeigt.
Partnersuchbegriff	Der in den Partner-Basisdaten angegebene Suchbegriff des Partners wird in dieser Spalte angezeigt.

Spalte	Erläuterung
Löschkennzeichen	In dieser Spalte wird angezeigt, ob bei dem gesuchten Partner ein Löschkennzeichen gesetzt wurde.
Zuständiger Mitarbeiter	Der zuständige Mitarbeiter laut Partner-Basisdaten wird angezeigt.
Landbezeichnung	In dieser Spalte wird die Landbezeichnung laut Partner-Basisdaten angezeigt.
PLZ	Die Postleitzahl des Partners wird in dieser Spalte angezeigt.
Partnerrolle	In dieser Spalte werden ggf. ein oder mehrere Icons angezeigt. Diese Icons verdeutlichen, welchen Partnerrollen der Partner zugeordnet ist. Folgende Partnerrollen werden ggf. angezeigt: • Kunde • Lieferant • Rechnungswesen • Mitarbeiter (nur bei Personen), • Kontakt • Ansprechpartner (nur bei Personen), • Vertreter • Spediteur • Marketing • Mitbewerber Zusätzlich können Sie auch erkennen, ob ein Partner bereits zum Löschen markiert ist. Das Icon ist in diesem Fall durchgestrichen. Siehe auch Liste der verwendeten Symbole und Icons im Bedienungsleitfaden.

4 Anwendungsbezogene Aktionen

Die folgenden anwendungsbezogenen Aktionen stehen in der Anwendung „Cockpit: Partner“ zur Verfügung:

- Aktion „Kontakte exportieren“
- Aktion „Kontakte importieren“
- Aktion „Synchronisieren mit SyncML Server“

4.1 Aktion „Kontakte exportieren“

Mit dieser Aktion exportieren Sie ausgewählte Partner in eine vCard oder CSV Datei. Diese Datei kann von verschiedenen Personal Information Manager weiter verwendet werden.

Nach dem Ausführen der Aktion öffnet sich ein Dialog-Fenster, in dem die folgenden Felder zur Verfügung stehen:

Feld	Erläuterung
Datei	Wählen Sie die Datei aus, in die Sie exportieren möchten. Geben Sie den vollständigen Pfad der Datei mit dem Schema „file:///“ oder „kstore://“ an. Vorschlagswert ist eine Datei im Standard-Arbeitsbereich der OLTP-Datenbank, an der Sie angemeldet sind.
Dateiformat	Wählen Sie ein Dateiformat bzw.- Dateityp. Mögliche Werte sind: • vCard Datei (Version 2.1) • CSV-Datei für Outlook (deutsch Version) • CSV-Datei für Outlook (englische Version)
Datei Zeichenkodierung	Wählen Sie eine Zeichenkodierung für die Ziel-Datei. Mögliche Werte sind: • Java Voreinstellung • US-ASCII • UTF-8 • Windows Latin-1 (windows-1252)

4.2 Aktion „Kontakte importieren“

Mit dieser Aktion importieren Sie ausgewählte Partner aus einer vCard oder CSV Datei. Diese Datei kann von verschiedenen Personal Information Manager stammen.

Nach dem Ausführen der Aktion öffnet sich ein Dialog-Fenster, in dem die folgenden Felder zur Verfügung stehen:

Feld	Erläuterung
Datei	Wählen Sie die Datei aus, die Sie importieren möchten. Geben Sie den vollständigen Pfad der Datei mit dem Schema „file:///“ oder „kstore://“ an. Vorschlagswert ist eine Datei im Standard-Arbeitsbereich der OLTP-Datenbank, an der Sie angemeldet sind.

Feld	Erläuterung
Dateiformat	Wählen Sie ein Dateiformat bzw. Dateityp. Mögliche Werte sind: • vCard Datei (Version 2.1) • vCard Datei (Version 3) • CSV-Datei für Outlook (deutsch Version) • CSV-Datei für Outlook (englische Version)
Datei Zeichenkodierung	Wählen Sie eine Zeichenkodierung für die Ziel-Datei. Mögliche Werte sind: • UTF-8 • Java Voreinstellung • US-ASCII • Windows Latin-1 (windows-1252)
Dublettenprüfung aktivieren (Checkbox)	Gibt an, ob beim Import auf mögliche Duplikate geprüft werden soll. Die möglichen Werte sind: • Checkbox deaktiviert: Prüfung findet nicht statt. • Checkbox aktiviert: Importierte Partner werden überprüft, ob passende Dubletten zu diesem existieren. Bei Verdacht auf ein Duplikat wird der entsprechende Partner nicht importiert. Der Vorschlagswert ist auf Dubletten zu prüfen. Die Einstellungen zur Dublettenprüfung werden im Customizing vorgenommen.
Alle Warnungen bestätigen (Checkbox)	Gibt an, wie Meldungen vom Typ „Warnung", die während dem Import auftreten behandelt werden sollen. Die möglichen Werte sind: • Checkbox deaktiviert: Warnungen werden nicht bestätigt. Meldungen vom Typ „Warnung" werden wie Meldungen vom Typ „Fehler" behandelt und führen dazu, dass die Daten nicht importiert werden. • Checkbox aktiviert: Alle Warnungen werden bestätigt. Meldungen vom Typ „Warnung" werden wie Meldungen vom Typ „Information" behandelt und beeinflussen den Import nicht. Der Vorschlagswert ist Warnungen nicht bestätigen.

4.3 Aktion „Synchronisieren mit SyncML Server"

Mit dieser Aktion übertragen Sie ausgewählte Partner zu einem SyncML Server. Es findet eine unidirektionale Synchronisation statt, das heißt die Daten werden nur zum SyncML Server übertragen, aber nicht umgekehrt.

Nach dem Ausführen der Aktion öffnet sich ein Dialog-Fenster, in dem die folgenden Felder zur Verfügung stehen:

Feld	Erläuterung
Customizing Einstellungen verwenden (Checkbox)	Mit dieser Checkbox legen Sie fest, ob die Einstellung, hinterlegt im Customizing, verwendet werden sollen. Wenn Sie die Checkbox deaktivieren können die Einstellungen übersteuert werden.
SyncML Server Adresse	Geben Sie die komplette HTTP- oder HTTPS-Adresse zum SyncML Server an, der synchronisiert werden soll. Das Feld ist nur eingabebereit, wenn Sie die Checkbox „Customizing Einstellungen verwenden" deaktiviert haben.
Benutzername	Geben Sie den Benutzername für die Anmeldung an den SyncML Server an. Das Feld ist nur eingabebereit, wenn Sie die Checkbox „Customizing Einstellungen verwenden" deaktiviert haben.
Passwort	Geben Sie das Passwort für die Anmeldung an den SyncML Server an. Das Feld ist nur eingabebereit, wenn Sie die Checkbox „Customizing Einstellungen verwenden" deaktiviert haben.
Kontakt Datenbank	Geben Sie den Datenbankname des SyncML Server an, auf der die Kontakte geführt werden. Das Feld ist nur eingabebereit, wenn Sie die Checkbox „Customizing Einstellungen verwenden" deaktiviert haben.

5 Customizing

Im Folgenden erfahren Sie, welche Einstellungen in der Anwendung „Customizing" für die Anwendung „Cockpit: Partner" relevant sind.

All relevanten Einstellung befinden sich in der Funktion „Beziehungs-Management"/Unterfunktion „Partner Synchronisation und Import/Export Einstellungen". Hier stehen die folgenden Felder zur Verfügung:

Feld	Erläuterung
Minimale Anzahl identischer Felder	Geben Sie die minimale Anzahl der Felder ein, die identisch sein müssen, um zwei verglichene Partner als Dubletten zu identifizieren.
Übereinstimmungsgrad	Geben Sie den Übereinstimmungsgrad an, den zwei verglichene Felder haben müssen, um als identisch identifiziert zu werden.
Nummernkreis	Tragen Sie den Nummernkreis ein, der bei der Nummernvergabe von importierten Partnern benutzt werden soll.
SyncML Server Adresse	Geben Sie die komplette HTTP- oder HTTPS-Adresse zum SyncML Server an, der synchronisiert werden soll.
Benutzername	Geben Sie den Benutzername für die Anmeldung an den SyncML Server an.
Passwort	Geben Sie das Passwort für die Anmeldung an den SyncML Server an.
Kontakt Datenbank	Geben Sie den Datenbankname des SyncML Server an, auf der die Kontakte geführt werden.
Proxy Server	Tragen Sie bei Bedarf den Proxy Server ein.
Proxy Port	Tragen Sie bei Bedarf den Proxy Port ein.

6 Business Entities

Für die Anwendung „Cockpit: Partner" ist das nachfolgende Business Entity relevant, das Sie beispielsweise verwenden, um

- Berechtigungen zu vergeben,
- Aktivitätsdefinitionen einzurichten oder
- Daten zu importieren oder zu exportieren.

Partner

`com.cisag.app.general.obj.Partner`

Das Business Entity gehört zu folgender Business-Entity-Gruppe:

`com.cisag.app.general.MasterData`

7 Berechtigungen

Berechtigungen können sowohl mithilfe der Berechtigungsrollen als auch durch die Zuordnung einer Organisation vergeben werden. Das Berechtigungskonzept können Sie in der Technischen Dokumentation „Berechtigungen" nachlesen.

7.1 Spezielle Fähigkeiten

Für die Anwendung „Cockpit: Partner" bestehen keine speziellen Fähigkeiten.

7.2 Organisations-Zuordnungen

Wenn die Funktion „Inhaltsbezogene Berechtigungen" in der Anwendung „Customizing" aktiviert ist, dann kann eine Person die Anwendung „Cockpit: Partner" nur nutzen, wenn ihr in den Partner-Stammdaten eine Organisation zugeordnet wurde, die mindestens in der folgenden Organisationsstruktur eingebunden ist:

- Vertrieb

7.3 Besonderheiten

Für die Anwendung „Cockpit: Partner" bestehen keine Besonderheiten.

7.4 Berechtigungen für Geschäftspartner

Die Anwendung „Cockpit: Partner" ist für Geschäftspartner nicht freigegeben.

Betriebswirtschaftliche Dokumentation

Vorgehensweisen: Erweiterung: Cockpit: Partner und Partner Synchronisation

Release	Ab Semiramis 5
Ausgabedatum	08/2004
Referenz auf andere Dokumente	Partner Cockpit: Partner Erweiterung: Cockpit: Partner und Partner Synchronisation

Inhaltsverzeichnis

1 Themenübersicht

In dieser Dokumentation werden die Vorgehensweisen für den Umgang mit der Partner Synchronisation Erweiterung der Anwendung „Cockpit: Partner" beschrieben. Die Vorgehensweisen enthalten Anleitungsschritte, z. B. wie Sie Partner mit Microsoft Outlook synchronisieren. Sie werden außerdem über mögliche Voraussetzungen und Auswirkungen informiert.

➢ Die Beschreibung der Anwendung „Cockpit: Partner", die unter anderem auch Feld- und Button-Beschreibungen enthält, finden Sie in der Dokumentation „Cockpit: Partner" und „Erweiterung: Cockpit: Partner und Partner Synchronisation".

2 Vorgehensweisen

Lesen Sie nachfolgend, wie Sie mit der Erweiterung der Anwendung „Cockpit: Partner" arbeiten:

- Partner mit Microsoft Outlook synchronisieren

2.1 Partner mit Microsoft Outlook synchronisieren

In dieser Vorgehensweise wird an einem Beispiel erläutert wie man Semiramis Partner nach Microsoft Outlook über SyncML synchronisiert.

Voraussetzungen

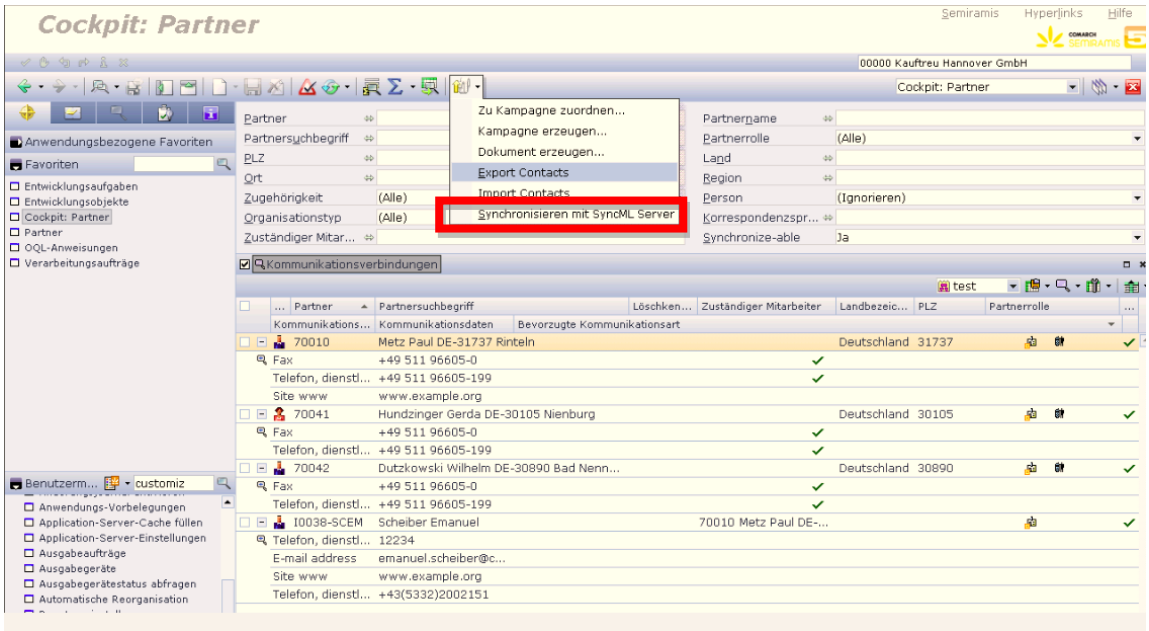
- Ein Benutzerkonto auf einem SyncML Server muss eingerichtet sein. Dieser muss vCard 2.1 kompatibel sein. Zum Beispiel bietet Funambol (http://funambol.com/) sowohl freie Server als auch freien Zugang zu ihrem Cloud SyncML Server an (https://my.funambol.com/).
- Falls ein Proxy Server und Port notwendig für die Verbindung zu diesem SyncML Server ist, müssen diese in den Customizing Einstellungen der Anwendung hinterlegt sein.
- Ein SyncML Client für Outlook muss installiert und eingerichtet sein. Zum Beispiel bietet Funambol einen kostenlosen Client an (https://outlook-client.forge.funambol.org/).

Anleitung

1. Öffnen Sie die Anwendung „Cockpit: Partner".
2. Schränken Sie die zu synchronisierenden Partner mit Hilfe des Abfragebereiches ein.
3. Drücken Sie dann in der Standard-Symbolleiste den Button „Aktualisieren".

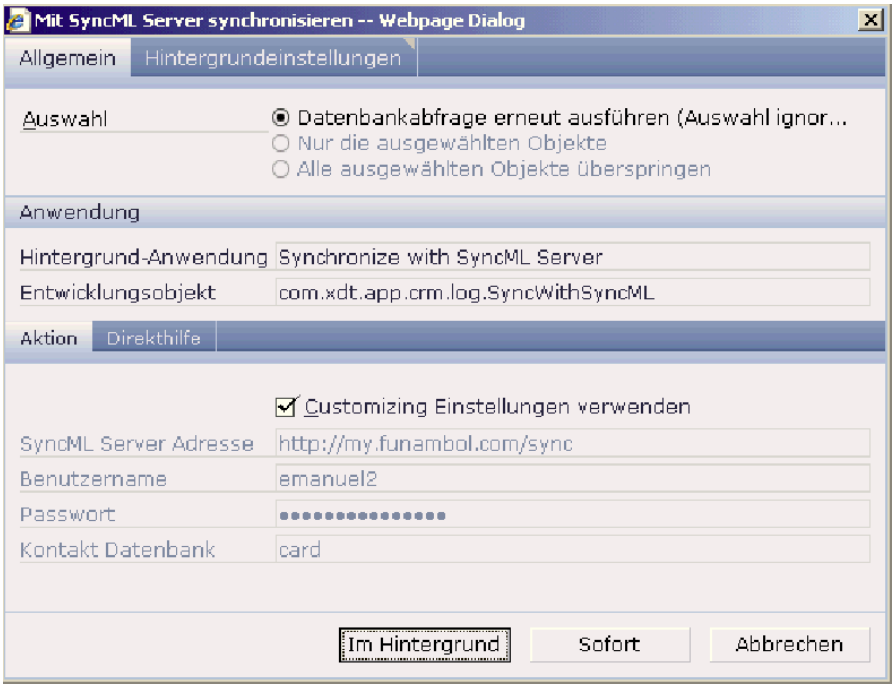
↳ Das System zeigt Ihnen die gefundenen Partner im Arbeitsbereich an.

4. Bei Bedarf können Sie die Auswahl der Partner unter Verwendung der Checkboxen weiter einschränken.
5. Wählen sie in der Aktionsrolle die Aktion „Synchronisieren mit SyncML Server".

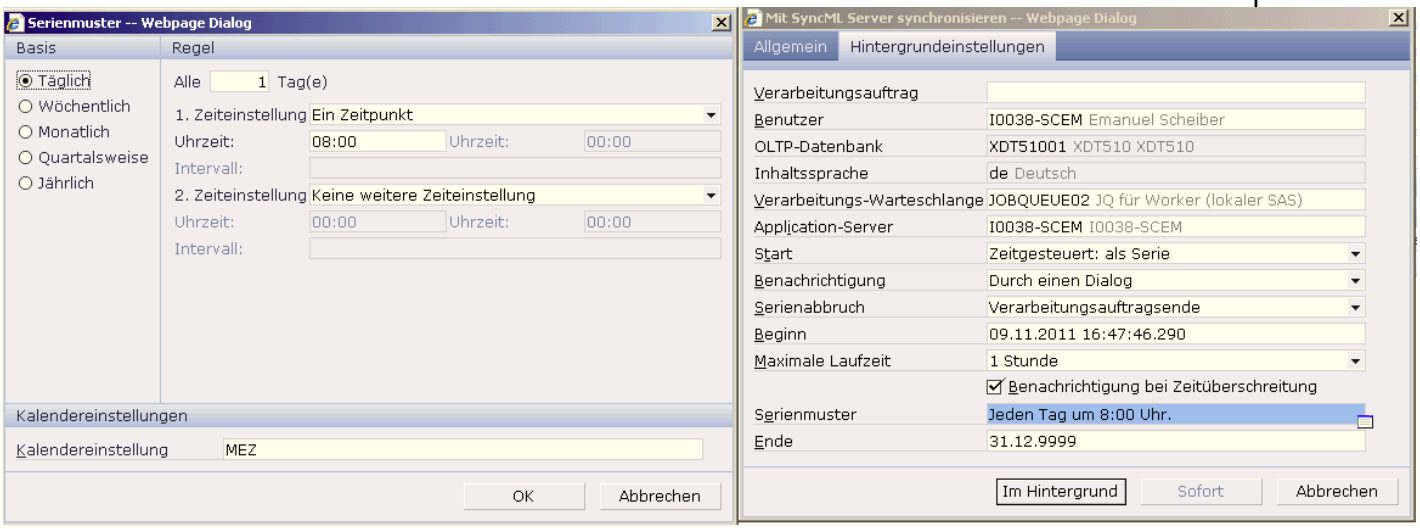


↳ Das System öffnet den entsprechenden Dialog. Falls Customizing Einstellungen vorhanden sind, werden die Aktion Felder mit diesen vorbelegt.

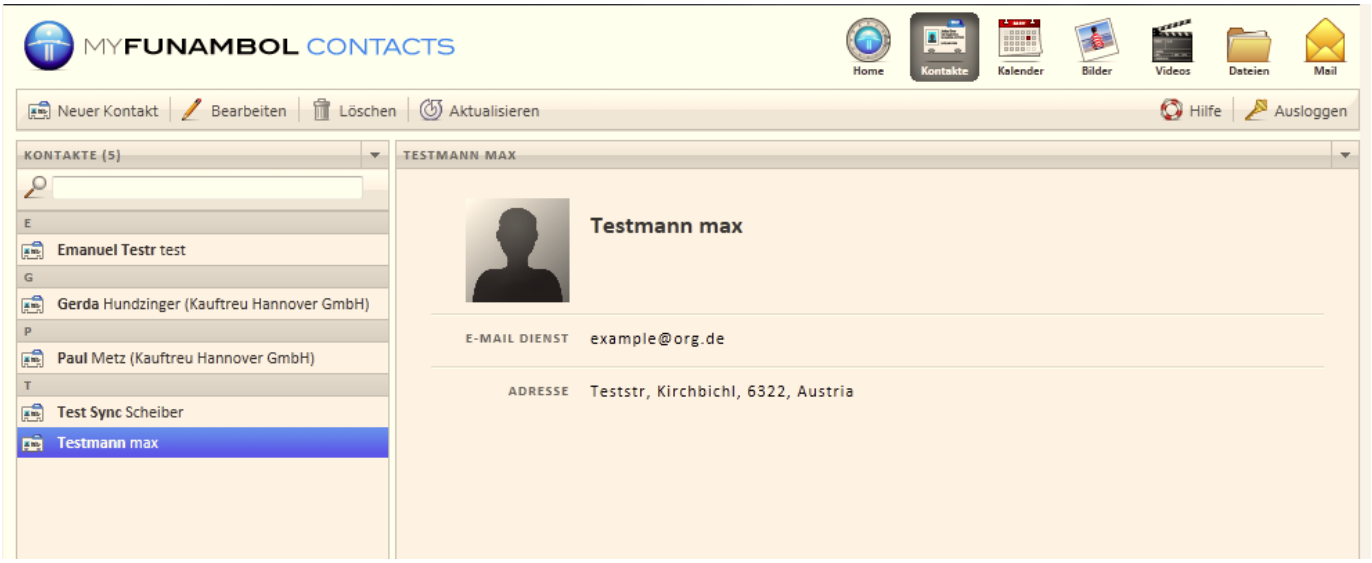
6. Sollten Sie noch keine Customizing Einstellungen hinterlegt haben oder wollen Sie diese überschreiben, dann deaktivieren Sie die Checkbox „Customizing Einstellungen verwenden" und befüllen Sie die entsprechenden Felder. Die SnyML Server Adresse für den Funambol Cloud Dienst ist „http://my.funambol.com/sync" und die Kontakt Datenbank ist „card".



7. Öffnen Sie den Tab Hintergrundeinstellungen, um Einstellungen für eine zeitgesteuerte Synchronisation anzulegen. Dafür wählen Sie für das Feld Start „Zeitgesteuert als Serie" und ein gewünschtes Serienmuster.
8. Falls Sie Hintergrundeinstellungen hinterlegt haben, drücken Sie den Button „Im Hintergrund" um die Eingabe zu bestätigen und die Aktion einzulasten. Falls Sie keine Hintergrundeinstellungen eingestellt haben, dann können Sie auch mit dem Button „Sofort" die Synchronisation augenblicklich einleiten.



9. Nun können Sie, falls Sie wünschen, am SyncML Server prüfen, ob die Partner erfolgreich übertragen worden sind.



10. Öffnen Sie nun Microsoft Outlook und den integrierten Funambol Client um die Kontakte mit dem SyncML Server zu synchronisieren.

↳ Die Kontakte wurden von Semiramis übertragen.