Framework for the Integration of Mobile Device Features in PLM

Zur Erlangung des akademischen Grades eines
Doktor der Ingenieurwissenschaften

von der Fakultät für Maschinenbau des
Karlsruher Institut für Technologie (KIT)

genehmigte
DISSERTATION
von

Jens Michael Hopf
Diplôme d'Ingénieur du Cnam, spécialité informatique
Grade de Master - Master's Degree

Hauptreferent: Prof. Dr. Dr.-Ing. Dr. h. c. Jivka Ovtcharova
Korreferent: Prof. Dr.-Ing. habil Ralph Stelzer
Datum der mündlichen Prüfung: 02. März 2016

Credits

Die vorliegende Forschungsarbeit entstand parallel zu meiner Tätigkeit bei der Dassault Systèmes Deutschland GmbH in Mainz. Ein herzlicher Dank geht daher an meinen Arbeitgeber für die wohlwollende organisatorische Unterstütz- ung. Meinem ehemaligen Vorgesetzten Herr Dr. Michael Grünert möchte ich meinen herzlichen Dank für konstruktive Gespräche und die fortwährende Unterstützung aussprechen. Ebenfalls möchte ich meinen Kollegen für wesentliche Inspirationen und fachliche Diskussionen danken. Die Entwicklung der Arbeit wurde auch durch eine Vielzahl von fachlichen Gesprächen auf wissenschaft- lichen Konferenzen begünstigt, die neue Betrachtungsweisen auf das Themen ermöglichten.

Mein besonderer Dank gilt meiner Doktormutter Frau Prof. Dr. Dr.-Ing. Dr. h. c. Jivka Ovtcharova für das Vertrauen, den Freiraum und die wertvolle Unterstützung sowie Förderung dieser Arbeit. Jede Lebenszyklusphase dieser Arbeit wurde von ihr intensiv und professionell mit kompetentem Rat betreut. Herrn Thomas Maier danke ich für konstruktive Diskussionen und Anmerkungen sowie die technische Unterstützung am KIT. Meinen Eltern danke ich für den ersten Personal Computer im Jahre 1995 und die wertvolle Unterstützung. Das Studium der anerkannten französischen Hochschule *Conservatoire National des Arts et Métiers* (CNAM) in Paris ermöglichte es mir, den Weg zur Promotion am KIT einzuschlagen. Daher geht ein herzlicher Dank an meine ehemaligen Professoren Prof. Dr. Inge Schestag, Prof. Dr. Peter Wollenweber, Prof. Dr. Johannes Arz, Prof. Dr. Alois Schütte und Prof. Dr. Edith Rost-Schaude.

Da so ziemlich jede Forschungsarbeit eine persönliche und intensive Herausforderung für einen selbst und für Freunde ist, möchte ich mich für die Unterstützung, Ruhe und Ausdauer meiner Freunde bedanken. Mein besonderer Dank gilt meiner Verlobten Julia Eymann für ihr Verständnis und ihre fortwäh- rende Unterstützung, die mir eine leistungsvolle Basis während der Jahre gege- ben hat. Weiterhin gilt mein Dank für zahlreiche inspirierende Diskussionen, Anregungen und kritischen Kommentare meinen langjährigen Freunden Raymond Vetter, Thorsten Schuh und Martin Böhme. Bedanken möchte ich mich auch bei Nicole Weidel und Dr. med. Anna Victoria Drüke für wertvolle Ratschläge und unterstützenden Beistand.

Weiterhin gilt mein Dank allen wissenschaftlichen Mitarbeitern und Studenten des Instituts für Informationsmanagement im Ingenieurwesen (IMI) am KIT, die bei der Implementierung des Frameworks durch wertvolle Diskussionsbeiträge mitgewirkt haben.

Karlsruhe, Dezember 2015

Jens Michael Hopf

Abstract

Mobile devices have evolved rapidly in recent years and have become an everyday commodity. This trend will increase in the coming years. The young mobile PLM market is still in its infancy. At present, companies have covered their business processes with stationary workstations while mobile business applications have limited relevance and have been used to a limited extent. Companies can cover their overall business processes more time-efficiently and cost-effectively when they integrate mobile users in workflows. Mobile features have the potential to significantly support business processes through contextual information and make them more effective. Moreover, entirely new workflows can be defined and established by considering mobile situations in business processes, which were excluded from the outset in the stationary context.

Today's mobile devices mostly provide access to PLM solutions via web browser. Mobile hardware capabilities such as GPS, camera, and sensor are scarcely used for business processes. For example, mobile users must manually enter the location name to perform a location-based search. Such context data could be determined on the mobile device and forwarded to the PDM system. The implementation of mobile business applications designed for different mobile platforms is time-consuming and expensive. At present, natively implemented business applications are designed, implemented, and distributed individually for each mobile platform and business process. This has the consequence that business applications have various communication modules and independently defined security procedures and mechanisms. Web-based business applications would solve these problems, but they provide only limited access to hardware features. So far there is no possibility to combine the respective advantage of native and web-based business applications for PLM, i.e. to model and manage various business processes centrally and to create dynamically new business processes considering mobile features. A bridge between the mobile world and the PLM world must be created in order to simplify business processes in the mobile context. Thereby the mobile user can be involved in various business processes without having to install and manage different mobile business applications. Moreover, the benefits of mobile features can be integrated into business processes. For this purpose, mobile devices have to be investigated regarding hardware characteristics and an overall model have to be designed and developed for integration into business processes. Mobile PLM applications for business processes are required to be modeled, centrally provided, and updated in context of the framework. Business processes can be simplified and presented more transparently because mobile users must not perform application updates on the mobile device. This ensures that mobile users always use the latest application version. Such situations of missing or invalid product information can be avoided by mobile solutions and also improve the productivity of the business because new updates are immediately available for other persons involved in the product lifecycle.

The objective is to develop not only a *Mobile Feature Framework* that can be used to model but also centrally control business applications for PLM processes using mobile features. The development of such mobile PLM business applications takes place in context of the framework and requires the featurization of the web-based logic. Mobile features are provided by the composition of mobile hardware and software logic on the mobile device. The framework allows the integration of mobile features in business processes of PDM systems a totally new and enhanced user experience through novel interaction and control components. All mobile users access mobile business applications and the underlying business processes through a mobile PLM client.

Kurzfassung

Mobile Endgeräte haben sich in den letzten Jahren rasant weiterentwickelt und sind zum alltäglichen Gebrauchsgegenstand geworden. Dieser Trend wird sich in den nächsten Jahren weiter verstärken. Der junge mobile PLM-Markt steht jedoch noch am Anfang. Bisher haben Unternehmen ihre Geschäftsprozesse stationär abgebildet und mobile Geschäftsapplikationen lediglich begrenzt eingesetzt. Durch eine vollwertige Integration mobiler Benutzer in Arbeitsabläufe können Unternehmen ihre Geschäftsprozesse zeit- und kosteneffizienter abbilden. Mobile Features haben das Potenzial, Geschäftsprozesse signifikant durch Kontextinformationen zu unterstützen und effektiver zu gestalten. Zudem können durch die Berücksichtigung mobiler Situationen in Geschäftsprozessen völlig neue Arbeitsabläufe definiert und etabliert werden, die im stationären Kontext ausgeschlossen wären.

Heutige mobile Endgeräte bieten größtenteils den Zugriff über den Webbrowser auf PLM-Lösungen an. Dabei werden mobile Hardwarefunktionalitäten wie GPS, Kamera und Sensorik der Geräte für Geschäftsprozesse kaum genutzt. Beispielsweise müssen mobile Benutzer ihren Standort selbst eingeben, um eine standortbezogende Suche durchführenden zu können. Dabei könnten solche Daten über das mobile Endgerät ermittelt und ans PDM-System weitergeleitet werden. Die Implementierung von mobilen Geschäftsapplikationen gestaltet sich durch unterschiedliche mobile Plattformen als enorm zeit- und kostenintensiv. Gegenwärtig werden nativ implementierte Geschäftsapplikationen für jede mobile Plattform einzeln pro Geschäftsprozess konzipiert, implementiert und verteilt. Dies hat zur Folge, dass Geschäftsapplikationen über verschiedene Kommunikationsmodule und

selbst definierte Sicherheitsverfahren und Mechanismen verfügen. Zwar würden webbasierte Geschäftsapplikationen das Problem lösen, dafür bieten diese nur eingeschränkten Zugriff auf hardwarespezifische Funktionen. Bisher gibt es keine Möglichkeit, jeweils die Vorteile von nativen und webbasierten Geschäftsapplikationen gemeinsam im PLM-Kontext zu nutzen, d.h. verschiedene Geschäftsprozesse mit Verwendung von mobilen Features zentral abzubilden, dynamisch zu modellieren und zu verwalten. Um Geschäftsprozesse im mobilen Kontext zu vereinfachen, muss eine Schnittstelle zwischen der mobilen Welt und der PLM-Welt geschaffen werden. Dadurch kann der mobile Anwender in verschiedene Geschäftsprozesse eingebunden werden, ohne dafür unterschiedliche mobile Geschäftsapplikationen installieren und verwalten zu müssen. Zudem sollen die Vorteile mobiler Funktionen in Geschäftsprozesse integriert werden. Dazu müssen mobile Endgeräte auf ihre Hardware-Eigenschaften untersucht und ein Gesamtmodell zur Integration in Geschäftsprozesse konzipiert und entwickelt werden. Mobile PLM-Applikationen für Geschäftsprozesse sollen dabei im Kontext eines Frameworks modelliert, zentral abgebildet und aktualisiert werden können. Geschäftsprozesse können so vereinfacht und transparenter abgebildet werden, da der mobile Anwender keine Applikationsaktualisierungen in Form von Updates auf seinem Endgerät durchführen muss. Dadurch wird gewährleistet, dass Anwender immer die aktuellste Version benutzen und dass Situationen mit fehlenden oder ungültigen Produktinformationen vermieden werden. Zudem wird die Produktivität des Unternehmens verbessert, da neue Updates für alle am Produktlebenszyklus beteiligten Personen sofort verfügbar sind.

Das Ziel ist die Entwicklung eines *Mobile Feature Frameworks*, mit dem Geschäftsapplikationen für PLM Prozesse unter Verwendung mobiler Features modelliert und zentral abbildet werden können. Die Entwicklung solcher mobiler PLM-Geschäftsapplikationen erfolgt im Kontext des Frameworks und erfordert die Featurisierung der webbasierten Logik. Mobile Features werden dabei durch die Komposition von mobiler Hardware und Software-Logik auf dem mobilen Endgerät bereitgestellt werden. Das Framework ermöglicht durch die Integration mobiler Features eine völlig neuartige und erweiterte Benutzererfahrung, da neue

Interaktions- und Steuerungselemente für Geschäftsprozesse in PDM-Systemen bereitgestellt werden. Alle mobilen Benutzer greifen über einen mobilen PLM-Zugriffsclient auf mobile Geschäftsapplikationen und die dahinterliegenden Geschäftsprozesse zu.

Contents

Appendix

List of Tables

List of Figures

List of Acronyms

AmI	Ambient Intelligence
API	Application Programming Interface
AR	Augmented Reality
ASP	Active Server Pages
ARMU	Augmented Reality Mock-Up
BOM	Bill Of Materials
BYOD	Bring Your Own Device
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CAM	Computer-Aided Manufacturing
CAx	Common term for Computer-Aided Technologies
CMMS	Computerized Maintenance Management System
cPDM	Collaborative Product Data Management
CPS	Cyber-physical Systems
CRM	Customer Relationship Management
CSCW	Computer-Supported Cooperative Work
DBMS	Database Management System

DDD Domain Driven Design

DIN Deutsches Institut für Normung (German Institute for Standardization)

DMU Digital Mock-Up

DSS Decision Support System

EAN European Article Number

EC Engineering Change

ECM Engineering Change Management

EDM Engineering Data Management

ERP Enterprise Resource Planning

EULA End-user license agreement

FDD Feature Driven Development

FfDF Federal Commissioner for Data Protection and Freedom of Information

IC Interactive Collaboration

ICT Information and Communication Technology

IDE Integrated Development Environment

IEEE Institute of Electrical and Electronics Engineers

IMEI International Mobile Station Equipment Identity

IoT Internet of Things

ISO International Organization for Standardization

JAX-RS Java API for RESTful Web Services

Java EE Java Platform, Enterprise Edition

JSON JavaScript Object Notation

GoF Gang of Four

GPRS General Packet Radio Service

GPS Global Positioning System

GUI Graphical User Interface

HCI Human Computer Interaction

HSDPA High-Speed Downlink Packet Access (3G)

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

LDAP Lightweight Directory Access Protocol

LTE Long Term Evolution

MD5 Message-Digest Algorithm 5

M2M Machine to Machine

MDA Mobile Device Application

MDC Mobile Device Client or Mobile Data Collection

MDF Mobile Device Framework

MDI Mobile Device Integration

MDM Mobile Device Manager

MDP Mobile Data Presentation

MDS Mobile Device Samples

MOM Message Oriented Middleware

MQTT Message Queue Telemetry Transport

NFC Near Field Communication

OOTB Out Of The Box

ORM Object-Relational Mapping

OS Operating System

QR Quick Response

P2P Peer to Peer

PnO People and Organization

PDM Product Data Management

PDP Product Development Process

PHP PHP: Hypertext Preprocessor / Personal Home Page Tools

PMU Physical Mock-Up

PLM Product Lifecycle Management

RFC Request for Comments

RFID Radio Frequency Identification

RUP Rational Unified Process

SCM Supply Chain Management

SDK Software Development Kit

SHA Secure Hash Algorithm

SLM Service Lifecycle Management

SOA Service-Oriented Architecture

SSL Secure Sockets Layer

SSO Single Sign-On

STL STereoLithography

TCP Transmission Control Protocol

TLS	Transport Layer Security
UML	Unified Modeling Language
UMTS	Universal Mobile Telecommunications System
URL	Uniform Resource Locator
UPC	Universal Product Code
UBE	User Behavior and Environment
UDP	User Datagram Protocol
VDMA	Verband Deutscher Maschinen- und Anlagenbau
VPDM	Virtual Product Definition Management
VR	Virtual Reality
VRML	Virtual Reality Modeling Language
WAP	Wireless Application Protocol
WSDL	Web Services Description Language
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WPAN	Wireless Personal Areas Network
WWAN	Wireless Wide Area Network
XML	Extensible Markup Language

1 Introduction

Your work is going to fill a large part of your life,
and the only way to be truly satisfied is to do what you believe is great work.
And the only way to do great work is to love what you do.

Steven Paul Jobs
co-founder, chairman, and CEO of Apple Inc. (1955 - 2011)

1.1 Description

1.1.1 Initial Situation

The number of variety of products on the market has increased significantly over the last few decades that consumers can easily find a wide range and assortment of goods. Nowadays, customers have better access to information resources related to consumer products than 20 years ago. Detailed product features, competitive market exchange, and consumers who share their experiences with others are no longer obstacles. Compared to earlier years, this results have been changed in several significant ways. The consumer behavior has been less continuous and unstable than before.

Consumer behavior has been changed in such a way that customers increasingly expect a product experience, while the product represents only a single component to meet the experience. An example of this shift represents the smartphone which can be purchased at low price, whereas many consumers buy the most expensive iPhone despite similar product features. Another example represents coffee capsules which cost three times higher than ground coffee or other prepackaged. The product experience is therefore strongly linked to the lifestyle in which consumers want to participate. It is equally important for the consumer to be able to identify with the product, because it is not only a product, but also offers valuable experiences. Through this shift, partly as a result of an improved market transparency, the power of the consumer has been expanded, so that companies must respond more to customer needs and thereby being found difficult to build a loyal consumer base who tends to switch faster the product and therefore the manufacturer. In order to ensure customer loyalty, the company must have a high level of innovation capability to place new ground-breaking products and solutions with unique characteristics on the market. Furthermore, it is essential to respond to customer demands for product features and services. The implementation of customer requirements even goes so far that such requests apply only to a small customer base even if deemed to be economically necessary. Due to varying consumption behavior as well as rapid technological changes, the expected useful life of the most products has been reduced significantly. For example, mobile phones are usually changed every two years into a newer model.

In order to increase competitiveness, companies must offer new products to the global market in even shorter intervals. A high pace of innovation and the constant shortening development cycles of products have become big challenges for companies and their employees. Whereas previously decisions were made only in the late phases of the life cycle, this has been postponed due to cost savings, shorten the product development, and production in an earlier phase of the design and construction process [Cf. ES09, p. 3]. Employees must be assisted by their company in their task to provide a permanent access to relevant information over the entire product life cycle at the suitable time in the right place and in the appropriate form where they are needed [Cf. Sen09, p. 16]. The quantity of information is not important, but rather a purposeful data provision for the respective (mobile) context of the person to fulfill and support them in their tasks. Overall, this also brings a benefit to the company.

Further, companies must carry out a closer integration of the various lifecycle stages of a product and have to encourage an enhanced collaboration and interaction of the people involved in the product lifecycle to meet the challenge. However, this implies a higher connection between the different stages of a product life cycle having a huge impact on other stages in a fault case. To reduce or avoid such effects, new opportunities for collaboration and interaction must be innovated and developed with the support of the different areas of product lifecycle. An improved collaboration with the appropriate areas could be achieved if a better supply of information takes place. When a timely and up to date data supply is established quickly, the areas can start to plan more effectively the next steps, which shorten the follow-up process and the product quality. For example, quality problems in the manufacturing phase of the product could be reported back from a mobile context to the planning and development stage to react immediately, solving inconveniences. It would be beneficial for other stages in the lifecycle to provide feedback immediately at the location of the newly created information back into the product lifecycle. The message is unlimited and can contain any type of information interacting between the PLM backbone[1] and the mobile context. In this work, this kind of information exchange is considered as *Mobile Information Context* (MIC).

1.1.2 Problem Statement

Companies are now using concepts such as product lifecycle management (PLM) to enhance and control the entire lifecycle of a product (see Figure 1.1) which manages all the phases of product design, from conception and engineering, to manufacturing, service and disposal. PLM integrates organizations, data structures, business processes, foreign systems, and suppliers to provide a repository for product information over its entire lifecycle. In 2010, a survey was carried out by Accenture, which has shown the challenges that companies will have over the next five years. These challenges still remain to be now more relevant than ever. The results demonstrated that the primary objectives in this study

[1] PLM-Backbone means a centralized repository of information

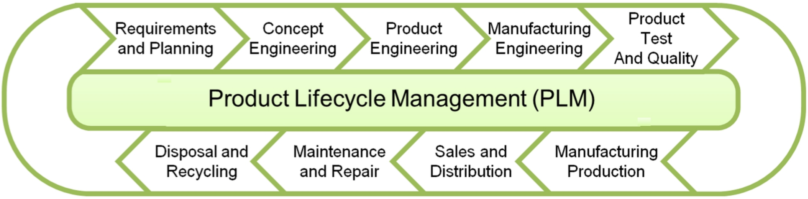


Figure 1.1: Generic lifecycle of a product

are to improve the innovation ability, engineering collaboration, and the reduction of complexity of product data [Acc10]. Especially in mobile situations, there are several ways to achieve these objectives. Most engineers today have a local workstation to manage either their products or the view of the production. These kinds of devices, which are laptops or personal computers have large bulky form factors, short battery life, and require long boot times. This daily business has significant limitations for engineers in a mobile context. Most of the values of PLM are not adapted quickly enough when engineers leave their local workstation. Thereby, most PLM solutions today have limited access to mobile situations. PLM requires a form of communication that allows collaboration across the boundaries of departments, whereas it has their limitations in providing information from a mobile context. While stationary workstations restrict the scope of availability for the PLM backbone, mobile devices can extend these boundaries. Figure 1.2 illustrates schematically the scope of conventional stationary device systems compared to mobile devices. Engineers could extend the capabilities of product innovation, social interaction, information exchange, and lifecycle acceleration when PLM has broken down the barriers of mobile limitations. Contribution of people in a mobile context requires more than an access to the web interface because mobility has their own eco system which assumes special demand such as display sizes, network coverage, security, and social aspects. Today's mobile apps provide mostly access to the PDM system[2] through the web browser. Mobile features such as location-based services, camera, sensors, Augmented Reality, and identification codes

[2] Product Data Management (PDM) System

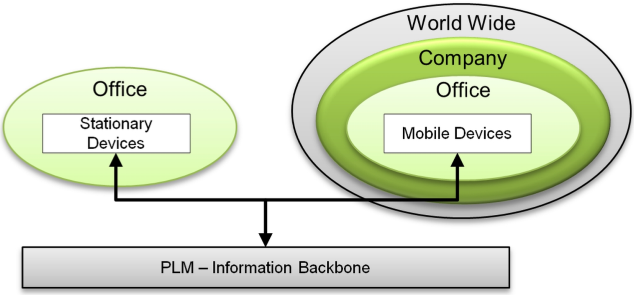


Figure 1.2: Scope of mobile devices

(e.g. as QR code[3] or barcode) are not integrated and therefore unusable for PDM systems. For example, mobile users have to enter their current location manually instead of using location-based services from mobile devices for search queries in PDM systems. However, those data on the device could be identified and disclosed to the PDM system. The diversity of mobile features is full of potential and opportunity for PLM. The capability in this area is currently unused because there are no mobile feature frameworks today that combine the advantages of mobile features with PLM.

In the field of Knowledge Management, research about user's context and presentation of content in the mobile environment has been already performed as determined in [EHTA11] and [BGT05]. Nevertheless, both approaches are limited to the semantic collaboration and content presentation. An approach for the integration of powerful hardware features in a generic way is still missing and is presented in this work. Methods focusing only on different types of mobile apps[4] have their limitations. The implementation of business applications (colloquially known as apps) designed for different operating systems like Apple's

[3] Quick Response (QR) code is a two-dimensional code developed by D. Wave in 1994

[4] Mobile application (colloquially known as a mobile app) is a software application mostly distributed through a application distribution platform and with a limited set of functionality which is designed to run on smartphones, tablet computers, and other mobile devices.

iOS[5] and Android[6] is very time-consuming and expensive, thus it is difficult to develop such apps as well as implement process changes in a short time interval. Today, those native business applications have to be individually designed and implemented for each mobile operating system and for each business process.

Moreover, the diversity of platform specific GUI[7] elements was examined in [LNF11]. The usability across all platforms for operational purposes on the basis of native applications is difficult. In a business environment it is important to access company-specific functions and processes using a unified way. Different user interfaces would negatively affect the productivity of the mobile user when they change the device platform. Figure 1.3 shows a GUI element comparison of the Phone Directory of the three most common mobile platforms which are Android, iOS, and Windows Phone. The usability across all platforms for operational purposes on the basis of native applications is difficult. Another aspect is the delivery of native business applications, because it is unusual to distribute company specific and internal software through application distribution platforms such as Google Play[8] or Apple's App Store[9] which are primarily designed for end-consumers. PLM applications are generally made for a selected circle of people who participate in the product lifecycle. Native business applications on mobile devices are characterized by a wide variety of platforms, and also by limitation of simultaneous updates on all mobile devices at the same time. Web applications (colloquially known as web apps) would solve the problem of updating, because they are browser-based largely executed on the server side, but they provide only limited access to hardware-features. So far, it can be hard to combine the respective advantages of native business apps and web apps together for PLM, i.e. to model and manage various

5 iOS (previously iPhone OS) is a mobile operating system developed and distributed by Apple Inc.
6 Android is a Linux-based operating system designed primarily for touch screen mobile devices developed by the Open Handset Alliance (Core member: Google Inc.)
7 Graphical User Interface (GUI)
8 Google Play, formerly known as the Android Market, is a cloud-based application distribution platform for Android developed and maintained by Google Inc.
9 Apple App Store is a digital application distribution platform for iOS developed and maintained by Apple Inc.

Figure 1.3: GUI comparison of the phone directory among the mobile platforms (adapted from [Cf. LNF11])

business processes centrally and to create dynamically new business processes considering mobile features.

The integration of mobile features extends the scope for PLM and enables the potential of mobile usage. Access to product information and processes on demand are beneficial for employees and machines. It is also crucial not only for people involved in different stages of a product lifecycle such as marketing, service, and maintenance, but also for people who need quick access to specific information depending on their context. Another group of mobile devices is represented by cyber-physical systems[10] that can collect information unattended

[10] Cyber-physical systems (CPS) are self-acting systems in a network with complex network structures that can adapt to changing conditions as needed. As an example, the smart grid is to be mentioned with the appropriate components such as the smart meter and electrical devices. CPS also describes the collaboration of software technical, mechanical and electronic elements in a system, which can interact with other units through the ability to communicate.

and communicate over a network infrastructure autonomously (e.g. sensor information such as air pressure or temperature). This group is characterized by a high complexity of the components, which may consist of software, mechanical, and electronic components. In context of companies, mobility can be used to support and optimize internal and cross-company values. It should particularly enhance efficiency and effectiveness of processes. This goal is achieved by a closer integration of mobile clients with the IT infrastructure supporting respective processing of value-added mobile tasks. Mobile clients are either employees of the company or end-devices/systems (robots). The classic mobile support for employees can be realized especially in areas where value added-related tasks need to be provided in an environment characterized by mobility [Cf. HFH+05, p. 9].

In addition, opportunities exist to increase the capability and efficacy through the integration between physical objects in the real world and business information systems [Cf. HFH+05, p. 9] to cover existing PLM workflows for mobile users. The overarching objective was already defined at a general and abstract level for the Internet of Things (IoT) in order to provide additional information for the realities and to minimize information gaps [Fle05, Cf.]. For PLM, this trend of linking information with the real world may become a new market if companies implement value-added parts of their business through business applications. This permits companies to handle their business processes in a cost-efficient manner and mobile features could make existing business processes more effective. Furthermore, even entire new business processes can be defined and established. To accomplish these objectives, a framework must be created in order to link the PLM information between the real and virtual world. Therefore, the aim of this research work is the development of a framework for PDM systems in order to reduce the information gap between the realities in mobile contexts. As a result, mobile features are now understood as core of the framework to supply the realities with additional information and to facilitate interactions.

1.1.3 Methodical Approach

A mobile feature framework must distinguish between three major points in triangle which depend on each other as follows: Firstly, the framework should focus on mobile users who interact in a mobile context. Secondly, it has to deal with the technical integration part which has a wide range of mobile operating systems and mobile features. Finally, the framework is dealing with different PLM providers with partly proprietary components. Figure 1.4 illustrates a generic representation of these study areas. The thesis is divided into different sections (see Figure 1.5):

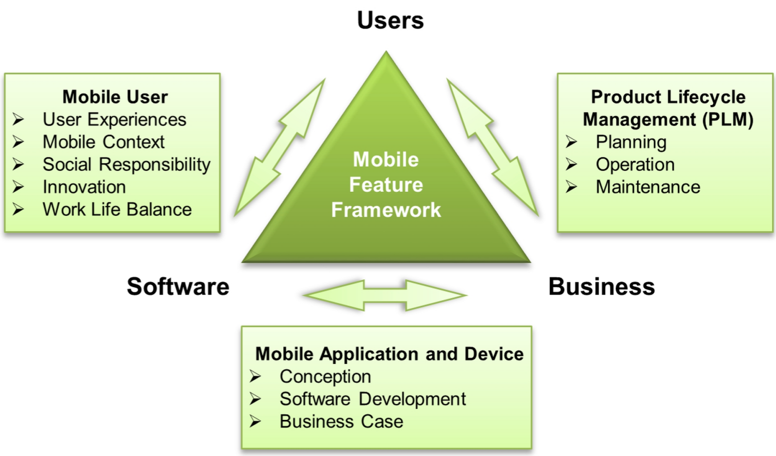


Figure 1.4: Study areas for the mobile feature integration

- Chapter 1 introduces the issue and determines the study areas which are divided into the sub-sections of mobile users and mobile experiences, mobile applications and devices used in a production environment, and parts of PLM related to a mobile context. The focus will be placed on the identified issues of the three areas.
- Chapter 2 analyzes and defines the social and mobile user requirements for businesses and the specifications for the information

technology such as infrastructure design and architecture for the Mobile Feature Framework. Subsequently, these dimensions are summarized and evaluated.
- Chapter 3 describes the general state of the art of concepts and techniques to identify existing gaps. Therefore, the chapter is divided into the sub-themes Mobile ICT and PLM and initially treated separately. Section Mobile ICT analyzes the behavior and culture of mobile users in order to investigate the mobile user context. In addition, mobile devices are classified and mobile platforms are studied. The PLM section define common terms, describes the functionality of PDM components as well as industries that have applied PLM concepts. Subsequently, both sections are considered together by examining the current design and engineering of PLM in the context of mobile ICT and mobile PLM applications. Frameworks in mobile contexts are treated separately.
- Chapter 4 deals with the overall domain model. First of all, the overall objectives of the framework are defined and an approach is represented in order to reach these goals. The approach is based on a methodology which is applied by the Mobile Feature Framework. Subsequently, the contexts and a ubiquitous language are defined. Furthermore, the extraction of patterns is performed based on user stories. The patterns and identified key issues are used to develop the entire domain model, which is then refined step by step by layering the architecture and defining components. In addition, the cross-component services and objects of the domain model as well as sub-models of the framework are presented.
- Chapter 5 determines and describes the groups of the most common native features of mobile devices such as file transfer, multimedia, and sensors. Subsequently, a model is presented and implemented by all features in order to ensure a unified basis for communication and processing within the framework. Therefore, the logical structure is defined for message styles, messaging systems, and communication channels as well as different stages of messaging such as construction, validation, and transformation. Finally, some mobile features are selected, planned, designed, and constructed.
- Chapter 6 covers the engineering process from the definition to the implementation of the framework as well as infrastructure

components. In addition, the persistence of data and framework structure from a developer's perspective will be discussed. The last part of this chapter is dedicated to framework testing which includes the exception handling.
- Chapter 7 deals with the validation of the Mobile Feature Framework. For that purpose, the validation method is selected and reasoned. Subsequently, the validation of the framework takes place by various case studies. The results of the implementation are evaluated and summarized.
- This work concludes by summarizing the key findings and an outlook in Chapter 8.

Figure 1.5 shows the structure of this research work. The design and implementation of framework components and features have been carried out by the author of this work without the involvement of third parties. However, the variety of inspirations was collected in technical discussions with industry partners who influenced the progress of the feature concept.

1.2 Outline of the Thesis

This section classifies the work and determines research areas which are derived and supported by studies. It starts by presenting different forms of mobile communication. This is followed by an investigation of mobile applications and mobile devices. In addition, an investigation is carried out to identify the roles PLM in this context. The end of the section summarizes the problem areas and gives an outlook of further actions. Figure 1.6 illustrates the research areas of this work.

1.2.1 Related Research Fields

Since the emphasis of this work is located on mobile ICT in context of PLM, no studies in the field of Virtual Reality (VR) are performed because VR requires stationary components and also concentrates mainly in the research field of stationary use. In contrast to Virtual Reality, in which

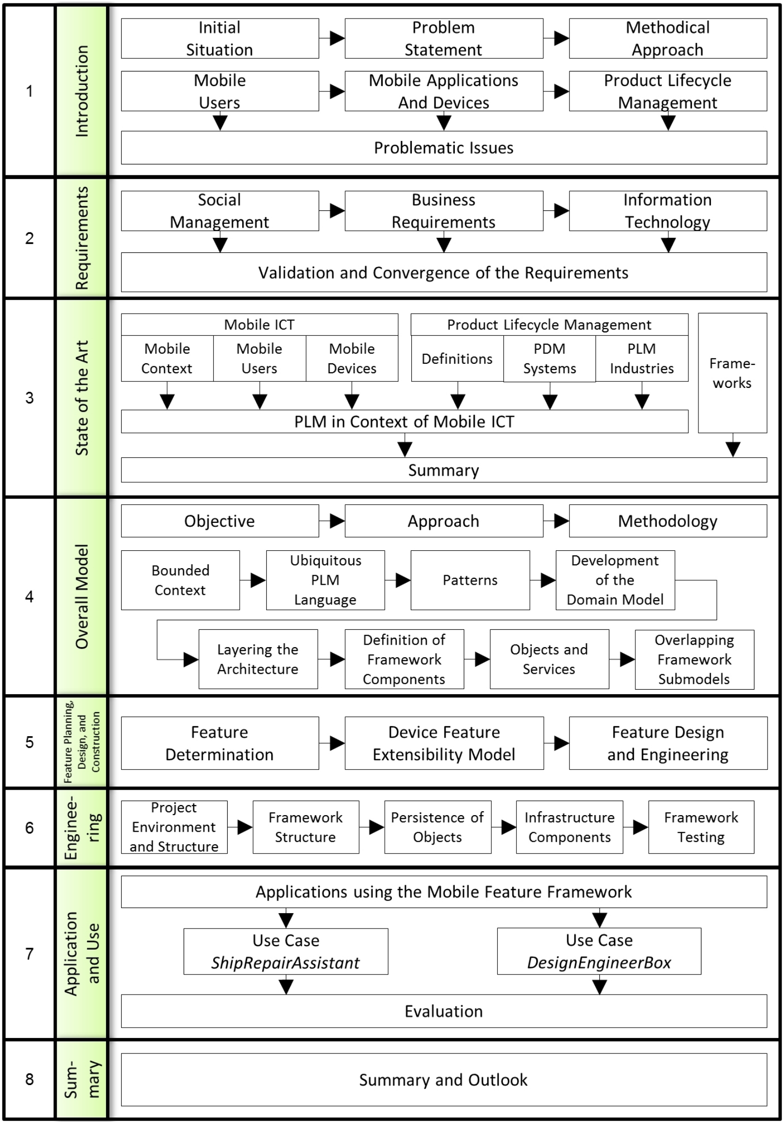


Figure 1.5: Structure of the work

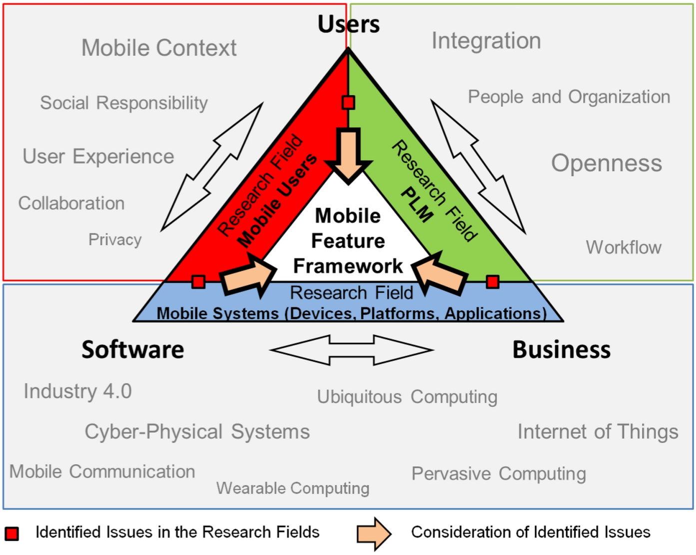


Figure 1.6: Determination of the study areas

users are completely immersed in a virtual environment, Augmented Reality (AR) is understood as the computer-based extension of the reality. A person's perception of reality is complemented through the visual presentation and interaction of information. The presentation of additional information can be achieved by multiple overlays to real images and videos. AR and VR represent sub-disciplines of *Mixed Reality*, which includes the entire *Reality-Virtuality (RV) continuum* [Cf. MTUK95, p. 282]. In recent years, AR has continued to gain in importance for the mobile sector, despite the use for commercial ICT solutions still occupies a low priority. In this work, AR is considered in terms of providing visual and interactive components for information. Also, an examination is carried out on the integration capability for existing AR software solutions as part of a mobile feature for the Mobile Feature Framework to make them available for new business process modeling in the context of PLM.

The term *Ubiquitous Computing*, introduced by Mark Weiser [Cf. Wei91, p. 94], refers to the approach of the ubiquity of information processing systems. In the view of Weiser, these systems are smallest networked computer systems which are not only characterized by a compact design, but also can communicate with each other and are integrated or docked into daily commodities. Those equipped objects with sensors are usually invisible to the user and can meet each task efficiently based on the context of use. The systems are usually developed for the appropriate condition and can respond to changing environmental conditions. This ambient intelligence is summarized under the general term *Ambient Intelligence*. The industry refers to the ubiquity of computers as *Pervasive Computing* under commercial use aspects, which are implemented with already existing technologies. For example, mobile technologies for entertainment and services are integrated into the vehicle and can be used with mobile features such as voice control and map services[11]. The totality of such networked and embedded computer systems is also referred as the *Internet of Things*. The term was used in 1999 by Ashton [Ash09] and aims to reduce the leak of information of different realities. In this work, the study of the mobile feature integration into the framework should create a bridge between the realities to minimize the information gap between the real world and virtual world. In this regard, an analysis and evaluation of mobile features is performed and the results are applied in form of identified pattern to the framework for the modeling of business processes. Due to the use of mobile ICT solutions, the mobile user is compulsorily influenced through pre-built tools for performing tasks. As a result, the mindset and actions of people may crucially changed in the social environment. In this context, the paradigm of *Persuasive Computing* refers to the computer-based influencing of human behavior. Mobile features should change the human behavior through the diversity of possible interactions to achieve a greater acceptance and penetration for the use of mobile ICT. This case considers as a particular challenge the

[11] Android car as part of the Open Automotive Alliance (OAA) with 28 automotive manufacturers and suppliers of mobile technology was announced on January 6, 2014 [Cf. Ope14]. Android Car apps running on the smartphone while the car provides preselected data and information are presented using the head unit of the car dashboard. Such apps and services can be only controlled over a standard interface of the Android Auto that does not distract the attention of driver from driving.

privacy and data security in terms of mobile users. Privacy is especially important when personal data are collected. For example, this can be biometric information obtained by portable computer systems. The development of such portable data processing systems is studied in the research field of *Wearable Computing*. Portable computer systems differ compared to other mobile systems because they are attached to the user's body during use. The aim of such systems is to support the user's activity as transparent as possible without restricting the execution of the activity or disturb the user. This work examines the feasibility of the portable computer systems integration as a further component of mobile features. In this regard, communication capabilities and protocols are analyzed and evaluated for data transmission.

1.2.2 Mobile Users

The research field of mobile users can be divided into three sub-areas. The technical level describes the challenges of the user dealing with mobile devices. The main focus is set on the user who has developed a corresponding behavior due to the technical circumstances. The social level represents the acceptance of the mobile user to communicate because of the technical capabilities of a device. Finally, the personal needs of the mobile user are investigated on the human level in order to achieve a high user acceptance of the mobile device.

- **Technical Level:** Mobile users differ significantly in comparison to stationary users. While stationary users are surrounded by a stable environment, mobile users are conditionally exposed by the mobility of constantly changing environments and external disturbances. Changing communication links with different levels of quality as well as device-specific limitations (e.g. display size and battery power), lead to changes in the behavior of mobile users. In this work, external disturbing factors are studied and suitable measures for the reductions are developed.
- **Social Level:** The social level deals with the communication behavior of mobile users. These examinations include communication mediums used for specific contexts as well as situations with social-related conflicts that cause a change of communication mediums.

Furthermore, the analysis of boundary conditions leads to a totally interruption of the communication and resumes only when a change in the current context takes place or boundary conditions are changing for the actual context.

- **Personnel Level:** The personnel level covers the consideration of personal needs including the topics of usability, user experience, data protection, and privacy as well as the ability to configure the device for personal needs. External factors also affect the behavior of mobile users in the interaction with the device. In this context, it is investigated how the integration of mobile features in PDM systems has a corresponding impact on these issues. The objective is to investigate if mobile features are generally accepted or rejected by users and which conditions must be fulfilled in order to achieve a broad acceptance rate of the user base.

1.2.3 Mobile Applications and Mobile Devices

In recent years, the number of mobile applications has grown rapidly on mobile devices such as smartphones and tablets. Mobile applications are dependent on the particular mobile platform, so that the development requires enormous efforts to ensure a consistent user experience across all platforms. The variety of mobile platform and device types represent the wide spectrum of mobile technologies which satisfy different uses depending on the context and can be used at various locations. This work investigates existing mobile features on conventional device classes and how these features can be abstracted and integrated for different business systems (e.g. PDM system). Another part of the research deals with the realization of present user interfaces for different mobile platforms. The key factors will be identified in order to ensure a consistent user experience on a variety of mobile platforms. Thereby, the development cost for mobile applications is to be reduced and provided in a shorter time period.

1.2.4 Product Lifecycle Management

Traditional worker's desk of engineers participating in the lifecycle, has mainly a local workstation in the office. This workstation is used to handle all engineering tasks and decisions from a single point. Once an engineer leaves the workspace, a contribution of new information is no longer possible and all important product data are no longer accessible if it has not been printed or noted. Therefore, product information can only be consumed unless it has been selected before. Any new generated information which is related to the lifecycle can only be transferred to the PLM backbone when the engineer returns to the workplace. Here, new knowledge could be lost when information has been only communicated verbally between the persons or written down on a piece of paper. In addition, new updates will be reported back to the lifecycle after a time delay. Other persons involved in the lifecycle could access the updated status only after a certain time. Incorrect product information can cause in such situations additional costs for the company and delays in the lifecycle. Such impacts could have a negative effect on the competitiveness of the company. In a study published by the Institute of Mechanical Engineering (ITM) at the Ruhr University Bochum in cooperation with IBM was identified as the primary potential benefits in the product lifecycle, the increase of the process efficiency through faster data distribution, data access, data search as well as less repeated re-entry of the same data [Cf. Abr09]. Such situations of missing or invalid product information can be avoided by mobile solutions and also improve the productivity of the business because new updates are immediately available for other persons involved in the product lifecycle.

In the field of CMMS[12], there are already approaches of a web-based communications between mobile devices and Maintenance Management System. Such approach is described in [CJP09] as a multi-tier architecture, which is based on CMMS and DSS[13] webservices. This paper has simulated a corrupt bearing, which sensor data was read from the machine and stored for data acquisition as well as signal analysis in the database. Mobile users remotely access the results through the

[12] Computerized Maintenance Management System (CMMS)
[13] Decision Support System (DSS)

middleware layer. In this context, a bridge between the field of PLM and CMMS/SLM[14] would be beneficial. A reasonable approach might not only provide the use of already stored specifications for a product (in this case for the roller bearing) in PLM for maintenance purposes, but also provide a feedback about potential failure of products from a machine back into the product lifecycle. Specifications and data about product service intervals could be retrieved directly from the PDM system and can be used for analysis and inspection machines by a mobile device.

In a research note by Gartner in 2012, some results were published regarding the transformation of the PLM software market by 2016. Thus, the use of 3D print for the manufacturing industries will play an important role. Companies will need to consider when evaluating the strategic implications of additive manufacturing instead of a traditional production. Additive manufacturing would have a huge impact on time, cost, and product quality, which would cause a serious change in the production and storage of spare parts. By 2016, at least 25% of discrete manufacturing will already have been accomplished by 3D printing to create product parts for production or services, which are used in the aerospace, heavy machinery, and industrial equipment industries, but only in areas which have no impacts on the structural integrity and product performance [Cf. HS12]. 3D modeling technology could play an important role for field service and spare parts supply company. A service engineer could analyze and identify a product defect with a mobile device and subsequently using the 3D print technology to produce a suitable part for replacement. This mobile context requires that the customer engineer must have access to internal company processes using the mobile device to get necessary information for the replacement, e.g. process instructions, product information and parts information for the 3D print.

[14] Service Lifecycle Management (SLM) is a concept that is linked to PLM and focuses on the life cycle of services. All service activities for planning, provision, and analysis of services are captured to optimize service processes such as customer support, service parts planning, and warranty management.

1.2.5 Identified Issues

- **Mobile Users:** The successful of this technology depends on the user acceptance of mobile information features. This requires an early analysis of mobile users in their behavior and subsequently their concerns will be taken into account. It should be noted that mobile users have not the same full-fledged work environment as stationary users and therefore a specific consideration of the respective context is necessary. Mobile users are vulnerable through their unpredictable behavior which makes it difficult to consider user's behavior for business applications. Moreover, mobile and social contexts are not sufficiently considered in the mobile communication these days which can lead to social conflicts within the contexts and can cause a negative impact on the related context.
- **Mobile Devices:** Mobile devices are characterized by their diversity and own property which are an enormous challenge for companies. There is no standard today that regulates the data exchange across all device classes so that each manufacturer has developed their own communication protocol. This makes it even more difficult to integrate different device classes into business processes without having to resort to the proprietary supplied software provided by the respective manufacturers. The manufacturer must consider appropriate standardized protocols. Likewise, different user interface components of mobile platforms will be a challenge for companies within an operational environment. Companies want to keep the provision costs of mobile applications for business tasks to a minimum and claim simple methods to flexibly adapt business processes.
- **Product Lifecycle in a Mobile Context:** The communication between the people involved within different stages of the product lifecycle runs today with a certain time break. There is no real-time update of a mobile context today because all information will be recorded in the product lifecycle with a time delay. Moreover, the consideration of mobile features does not take place in mobile PLM applications. Through a better integration of mobile users, this gap could be filled when information are accessible in a timely

manner as soon as they arise. For example, in a meeting room or in the production hall, the respective place of origin information is important in this context. Furthermore, additional information generated on mobile devices could be automatically inserted into the lifecycle. For example, such data might be data logger or measurement devices.

2 Requirements

Chapter 2 analyzes the requirements of the framework for the integration of mobile features. The requirements are obtained from different views. The following perspectives are considered: sociological management, the business requirements for mobile contexts with respect to PLM as well as the perspective of information technology with regard to the Mobile Feature Framework.

2.1 Social Management

This section discusses the social aspects of mobile users and is divided into the following three areas: privacy policy, user context and behavior as well as mobile user culture. The first part determines the requirements from the perspective of the mobile user and the company. The second part defines the requirements of user's context as well as user's behavior in certain situations. The final section deals with the cultural aspects of mobile users.

2.1.1 Privacy Policy

The Privacy Policy is divided into two areas. The first area focuses on the risks arising from the company during the transmission of data to mobile devices. The second area deals with the risks of the total transparency of mobile users through the surveillance by companies.

Data protection risks for companies

The smartphones as part of the mobile device groups carry a high security risk for companies through a large number of third-party apps. Apps have significant deficiencies in dealing with user data and the transparency with respect to the mobile user. In a study by the *Pew Internet Project* about *Privacy and Data Management on Mobile Devices* was determined that 57 percent of mobile application users have concerns about their privacy reasoned by collecting and sharing their personalized user data [Cf. BSM12, p. 2]. In a survey conducted by TRUSTe regarding privacy was concluded that 74 percent of the customers believe that it is important to know which personal information is collected by mobile applications, while 98 percent want easily access the privacy controls [Cf. TRU11, pp. 5–13]. The type of personal information that is processed and transmitted to third parties after the installation of non-business apps is mostly uncertain. The transmission of company specific and sensitive data is fraught with risk because competing companies could collect and use information for their own competitive advantages. 90 percent of the employees have been permitted to retrieve company data, while 41 percent have access to sensitive data [Cf. Hen12, p. 1]. Table 2.1 shows major security concerns on mobile devices in respect of data storage and also gives an overview of which requirements are derived for the Mobile Feature Framework.

Privacy risks for mobile users through companies

PLM data such as print drawings or 3D models represent confidential business information and require adequate protection mechanisms to ensure the safety supported by today's standards in mobile technologies. This may result in risks for mobile users because the security must be paid with an increased transparency of the users. Therefore, it is important to find the right balance between the transparency of mobile users and the protection of business data. It would be appropriate to enable tracking capabilities of the mobile device only when it is needed to fulfill a task. Furthermore, the mobile user should be informed when a function is using mobile features to carry out other activities.

Risk	Hazard Type	Framework Requirement
⇑	Employees with travel activities	Restricted access to sensitive business data
⇑	Employees whose mobile device was stolen or is lost	Deactivate user accounts to prevent data requests, delete company data after a defined period of inactivity, and track activities
⇒	Forwarding sensitive corporate data with private mailing or messaging services	Allow data exchange only for certified applications
⇒	Copying content from the company's app into third-party apps	Disable or block clipboard feature
⇒	Saving corporate data in unprotected areas on the device	Encryption and storage of encrypted data
⇒	Data transfer to uncontrolled external storage	Block external data storage when saving data
⇒	Screen sharing and capturing applications	Deactivate user accounts and report screen sharing activities
⇒	Accessing shared components such as phone contacts	Use separate contact lists with restricted access for third-party apps
⇒	Retired employees	Deactivate user accounts and delete company data from mobile devices
⇒	Data access outside the working hours	Restricted access to PLM data outside the working hours

Table 2.1: Security concerns for companies on mobile devices

The tracking of the mobile user should be carried out only within the working hours. The 24-hour tracking of the mobile user would cause privacy concerns[1] and therefore should generally be avoided. Table 2.2 presents concerns from the perspective of mobile users and derives the requirements for the Mobile Feature Framework. The separation between the business and personal use of mobile devices is not completely possible and requires a separation of personal and business data. In a report from Ovum in 2011 was determined that less than 23 percent of the IT workers believed that employees using personal devices for work activities could have more business advantages, which preponderates the hazards for companies [Cf. Kel11]. This is related to the IT staff which has a deeper understanding about security needs of the company compared to employees of other departments. Therefore, companies sometimes provide smartphones which are used only for business purposes. An alternative represents *Bring your own device* (BYOD) where employees use their private mobile device for business reasons. However, this access is usually limited to few company services, because legal questions about data privacy are still unanswered and only limited personal data as well as operating and business secrets are to be processed in partially uncontrolled mobile environments.

Risk	Hazard Type	Framework Requirement
⇑	24-hours tracking of the mobile user carried by the company	Limited temporal tracking, only used to perform a task
⇒	Retrieve status information provided by the mobile user	The usage is determined by the mobile user itself
⇒	Sharing and transmission of information captured from private situations / contexts	Context data is forwarded / shared and has be determined by the mobile user

Table 2.2: User concerns while using mobile devices

[1] According to the Federal Data Protection Act (FfDF) in Germany, the covert tracking is inadmissible because it is understood as “Mobile personal storage and processing media” within the meaning of § 3(10) and § 6c. For a mobile device with fixed assignment to the user, the regulations § 4, 32 of the FfDF apply. Thereby, a total surveillance of users is disproportionate. The recognizable and limited localization of users for specific purposes is permitted in compliance with the legal regulations. These paragraphs are not applicable for mobile devices without user assignment.

2.1.2 User Context and Behavior

Mobile devices are able to recognize external influences (e.g. automatic brightness control for touch screens), but mobile features are not sufficiently taken into account in software projects for business use. The inclusion of the user behavior in mobile business applications takes place only partially. However, user context and behavior play an important role when it comes to enhancing the user experience. Mobile users address requirements to the mobile framework to ensure that external influences are considered in PLM applications which are provided by the framework. External influences of mobile users must be taken into account in PLM applications which are provided centrally through functions of the Mobile Feature Framework. Commonly behavioral patterns such as muting the phone during a meeting would signal other users that a disturbance of the mobile user in the current context is unfavorable. Table 2.3 lists some examples of requirements for user behaviors and contexts.

Context	Framework Requirement
User Behavior and Context	
Activating mute function on mobile device	Blocking requests or information delivery in silent mode
Repeated data retrievals of certain information	Content caching and offline capabilities
Device Context	
Location-based search queries (e.g. search queries for people and parts)	Using GPS or mobile networks for search queries (e.g. show only people near my location)
Object navigation (e.g. people, parts)	GPS functionality combined with PLM information (e.g. navigation to the object based on coordinates)

Table 2.3: Mobile user behavior and context with related requirements

Collected personal data are processed only for actual and context-related user tasks. Such measures should help to ensure that user's personal data cannot be misused for context-foreign applications. The user must be informed if a data collection is performed on the mobile device. Moreover, the mobile user should be given the possibility of a client-based control by agreeing to the collection and transmission of data.

2.1.3 Culture

The behavior of the mobile user is deeply rooted into the culture and linked to social aspects. Cultural aspects are often not sufficiently taken into account in mobile software. Sometimes the software is adapted for specific regions and sold exclusively if it is economically useful. The perception of cultural backgrounds is complex, because the patterns differ in a high degree from culture to culture. Thereby, the privacy of every culture is more or less pronounced. Cultural aspects are comprehensive and diverse. Therefore, a more detailed and country-specific research that captures the relevant cultural aspects are required. The demand that can be derived for the mobile framework is a consideration of cultural aspects for a flexible integration of future culturally related to features.

2.2 Business Requirements for mobile PLM

The business requirements for the mobile framework are derived from the different perspectives that provide a valuable contribution to support the product development. The perspectives of the product development, the engineer, process model, and mobile device as well as the product data management system are considered.

2.2.1 Engineers

The access to mobile features through mobile PLM applications must be made for the Engineer in an intuitive way. The Engineer must have an added value without significant additional workload. The design of the user interface needs to be adapted for mobile situations to avoid an unpleasant user experience. The requirements addressed to mobile PLM applications are device-independent interaction between the mobile user and the value-added PLM service. Any user interaction may not be designed for a particular device class. The user interface must also take the latest technical standards into account. Since mobile devices have smaller screen sizes than desktop PCs, the operability must be adapted for simple UI structures. Flat and easy navigations as well as reduced information structures should be preferred. The consideration of established usability conventions in relation to the mobile usage context must be observed. The requirements from the perspective of the engineer are addressed to the consideration of personal needs as well as the adjustment to social and situational contexts. Therefore, the user experience for mobile web design is divided into three dimensions, which is summarized in Table 2.4.

Dimension	Framework Requirement
User	Fulfillment of needs of the mobile user and consideration of user's environment
Mobile Device	Device-independent usage of PLM applications
Content	Simple navigation structures and adapted content in PLM applications

Table 2.4: Engineers view on requirements (adapted from [Cf. Roz09])

2.2.2 Process Model

The phases of the product life cycle are examined for patterns, subsequently analyzed and abstracted. The results represent the base of the framework design. The pattern search concentrates on PLM areas that meet the following criteria:

- Processes and workflows that show a high value for mobile users
- Highly frequented scenarios by potential mobile users
- Promote innovative capability
- Visualization of ideas and any kind of information
- Access to PLM information needed in mobile contexts
- Share content with other people in a quick and accelerated way
- Capturing information that arises in mobile contexts

The provision of mobile value added services through PLM applications may not result in a proportional or progressive increase of the workload for the user. Therefore, the framework requirement is to examine the economic benefit of the mobile value-added services and to consider only patterns in the framework, which provides a high added value. An easy integration into existing processes and procedures should be feasible. In addition, the framework should provide the support to delegate tasks of the user between stationary and mobile devices. Moreover, an approach during the modeling of the framework should consider the capabilities to pause and resume already partially started task and to complete at a later time. Thereby, the user should be able to pay more attention to the surrounding environment. Necessary task interruptions must run smoothly and with low transition time for the initialization to avoid a negative effect on the user experience. The framework should have a modular design that allows an adaptation for dynamically changing conditions to afford the individual requirements of mobile users. Today's proprietary mobile solutions offer specific services without mobile features and they do not quite support to the respective mobile context. Thus, current mobile PLM applications are inflexible and uniformly for all mobile users. Mobile features should be usable for mobile PLM applications provided by the Mobile Feature Framework. In addition, mobile contexts should be determined by Mobile Feature Framework in order to adapt flexibly to the particular mobile user. Based on the collected information by the framework, the user interaction and UI-forms of mobile PLM applications should be able to change dynamically. This kind of mobile user profiling based on the mobile context pursues the goal to offer exactly the information that correspond to the current context and match interests of the user. To ensure that all context-related data is available, the mobile PLM application must be used for different areas of the product's life cycle.

2.2.3 Mobile Devices

Mobile devices must be examined for their ability to communicate with other systems and need to be classified. This also represents a requirement to the framework whose results are considered in the modeling of the communication protocol and the data exchange format. Mobile PLM applications can only use respective mobile features provided by the framework that are supported by the mobile device. Mobile devices should have the ability to improve the performance of a mobile communication, as well as components for controlling mobile features, such as measuring and sensor technology.

2.2.4 Product Data Management Systems

The demands which are addressed to PDM systems are the usage of unified and standardized methods for data exchanges. Mobile PLM applications should be able not only to obtain and store information in PDM systems, but also to trigger the data processing. The data exchange should base on an open and standardized protocol to ensure the interoperability.

2.3 Information Technology

The section information technology covers requirements for the IT infrastructure as well as the Mobile Feature Framework for the following topics: IT security with respect to the data flows within the IT infrastructure; Mobile networks which are mainly used by mobile devices; openness and interoperability of PDM systems and mobile platforms; the ability to integrate mobile features; expandability, adaptability and reusability of PDM systems and other essential framework requirements as well as applications using the framework.

2.3.1 IT Security

PLM is for many companies the source of revenue because product innovation and knowledge are stored in a form of documents and drawings in the PDM system. Security plays an important role in the PLM environment to protect the intellectual property of the company. An industry study carried out by the *Verband Deutscher Maschinen- und*

Anlagenbau (VDMA) in 2012 regarding product piracy, showed that the theft of intellectual property through counterfeit products has caused an enormous economic damage for companies. Thus, the revenue loss in 2011 was estimated at 7.9 billion euros, which means a 24 percent increase compared to 2009 [Cf. Ste12, p. 5]. Therefore, the framework requirements are necessary to protect the intellectual property through appropriate measures. Information security pursues three primary objectives of confidentiality, availability, and integrity of data [Cf. Bun12, p. 14]. These objectives are discussed in Section 2.3.1 and supplemented through additional requirements. The areas are subdivided into the following sections: Confidentiality, authentication and authorization, data encryption on device, data integrity, and remote security.

- **Confidentiality:** Data can only be accessed by authorized users. Data access without an authorization must be detected and blocked. In addition, a mechanism must control the disclosure and publication of data for authorized users. The access to information must be imaged by a set of rules, so that a modification to the rules influences the access to the information.
- **Authentication and Authorization:** The term authentication is understood to verify the user's identity. This procedure is usually supported through protected user knowledge, for example by a user name and password. Since the password is set by the user, it can be cracked in a short time depending on the complexity. Methods such as brute force attack[2] or dictionary attack[3] helps to accelerate the password cracking. Password rules can increase the complexity of an identification and also the safety, but every precaution is torpedoed when the user saves the password in a notepad or will be observed during entry. This method to verify the user's identity has shown weaknesses in the past [Cf. DHA10], therefore other methods must be considered. A study conducted by Bonneau has analyzed and compared over 70 million user

[2] Brute force or exhaustion is a method which use all solutions applied to a problem until one successfully matches. Because this simple approach is performed to find a solution using raw force, there is only few logic for an implementation necessary.

[3] This method is guessing the password by repeated trial and error using a dictionary. The dictionary contains a list of strings in random order or in decreasing order of probability of selection[Cf. Bis03, p. 312]

password hashs[4] regarding password strength. Only few complex password variations were discovered in the datasets while identifiable user groups have used comparably weak passwords. The user's motivation to use complex passwords was less dependent on the system or demographic factors such as age and nationality [Cf. Bon12]. Because mobile devices such as smart phones are personalized, device features can be used to perform an advanced identity verification through mobile features. For example, device features are characteristics by IMEI[5] number or network provider. The combination of device features with identity verification would increase the security for mobile data accesses. The location of the mobile device could be detected through GPS[6] and WLAN to limit the access to PLM information. Sales representatives who travel to foreign countries would have less access privileges to sensitive data than traveling in their home country. Figure 2.1 shows an example for advanced identity verification through a mobile feature.

- **Data Encryption on Device:** The handling of PLM data on mobile devices represents another important aspect. Configuration data must be stored for native applications as well as PLM data such as design files and descriptions need to be loaded from the PLM environment for editing and viewing tasks. Sensitive information must be stored in a secured data area to ensure the protection of the corporate data against unauthorized access. The secured data area should be accessible only for authenticated and authorized users (see Section 2.3.1). Unneeded information should be deleted irrevocably from the secure data area to minimize the risk of unauthorized data access.
- **Data Integrity:** The integrity of data must be guaranteed to ensure that unauthorized manipulation of a source is precluded. A message must be sent to the same recipient in the same manner

[4] A hash function is using an algorithm (e.g. SHA-1, MD5) to encode a plain string into an encoded string. It is a one-way function without the possibility of reversal or recovery the original string)

[5] The International Mobile Station Equipment Identity (IMEI) is an normally unique 15-digit numeric number assigned to each individual mobile station equipment for identification purposes[Cf. 3GP09]

[6] Global Positioning System (GPS)

as this has been transmitted by the sender [Cf. Bun12, p. 14]. Nevertheless, if messages contains undesired and unexpected modifications, this should at least be recognized by the mobile feature framework.

- **Remote Security:** Mobile devices can have their location of use outside the company. This implies that the access to PLM data can occur from a physical environment that appears to be less trustworthy. Therefore, special privacy protections must be in place to control the access to the mobile device. A locking and erase mechanism should exist in case of device loss to irrevocably delete sensitive data from the mobile device and to prevent abuse.

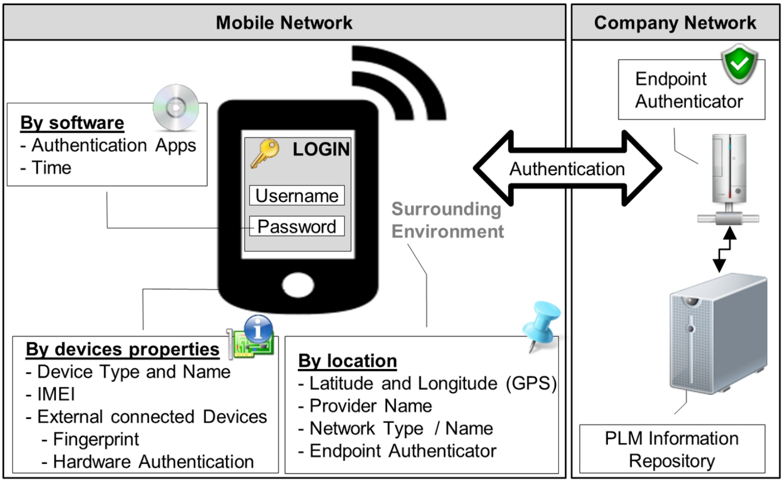


Figure 2.1: Advanced identity verification through mobile features

2.3.2 Mobile Network

Mobile networks are subdivided into five areas to analyze the requirements for the framework: Network reliability, system availability, standardization, costs for transport of mobile data, as well as encryption of communications.

- **Network Reliability:** In general, network protocols are used for wired networks and mobile networks such as GPRS[7], HSDPA[8] and LTE[9]. Thereby, the reliability is an essential factor for the usage of network protocols in mobile environments. The accuracy of mobile networks differs greatly compared to wired networks. Software that operates in mobile environments must be tailored to consider network interferences such as interruptions and slow transfer rates. Table B.1 in Appendix B.1 compares the differences between stationary and mobile networks.
- **System Availability:** The availability should be considered from three perspectives: the mobile devices, the feature framework as well as the PLM environments. Mobile devices are exposed to various influences which means that a permanently data connection cannot be alive. It may happen that there is no reliable network coverage available, the battery of the mobile device reaches a critical level, or the mobile user is busy with other activities. The second view refers to PLM environments that must meet the initiated requests from mobile devices within a predefined period of time. System failures must be prevented by appropriate backup solutions, because the availability is an important quality criterion for the overall system architecture. The third perspective from the framework must manage and translate the requests from the mobile devices and PLM environments. It represents the key module of the system architecture and a connector between both areas.
- **Standardization:** The framework has to use standardized protocols and third-party systems to ensure the reusability of technologies. Proprietary software should be avoided, because the changeability usually depends on the license agreement which cannot be guaranteed over a longer time period.
- **Costs for Transport of Mobile Data:** The cost of transporting data to the mobile devices depends on the network provider and can vary enormously. Through suitable compression method, data can be compressed to reduce the amount of transmitted information.

[7] General Packet Radio Service (GPRS) (2G and 3G)
[8] High-Speed Downlink Packet Access (HSDPA) (enhanced 3G)
[9] Long Term Evolution (LTE) (4G)

The retrieval of large data sets across mobile networks should be designed in a way that allows mobile users to receive data on demand when this is desired.

- **Encryption of Communications:** According to the IT security, it is required to encrypt data transmission to mobile devices in order to protect sensitive product information outside the company. Because mobile devices are connecting to mobile or wireless networks, they are exposed to the risks of sniffers, which may record the connection between the mobile device and the access point. Passwords, drawings, and other data types could be exploited. For this purpose, network protocols with secure data transmission, such as Transport Layer Security (TLS) have been developed. TLS is known by its former name Secure Sockets Layer (SSL) and is used with a variety of encryption methods in all current web browsers and many applications. Therefore, the need for encrypted data transfers is an important framework requirement.

2.3.3 Openness and Interoperability

The requirement for the interoperability of the framework is determined by the ability to communicate with different systems and technologies. The term is often used in various definitions by IEEE[10] in [Cf. IEE90, p. 113], by Chen in [Cf. CDV08, p. 648], and compiled in a comparison by Kosanke in [Cf. Kos06]. The framework applies the following own definition:

> "Interoperability means the cooperation and openness in relation to platform and network-independent communication in order to exchange and process information in every conceivable way."

The framework must ensure the communication to different mobile device classes and PLM environments based on standard technologies. Proprietary technologies have to be avoided, because they are not open for everybody and not able to establish over a longer time on the market.

[10] Institute of Electrical and Electronics Engineers (IEEE) is a professional association of engineers for electrical and information technology

Furthermore, there is a risk of dependence from a software provider as well as an inadequate support of the industry by a lack of standards. In order to establish a widespread acceptance of the framework, open standards are necessary. Therefore, the use of standardized protocols and open data formats has to be favored. The openness of the framework represents a requirement to ensure interoperability.

2.3.4 Integration Capability

To meet the increasingly rapid market changes, new requirements are placed on the integration capability by customers. Integrations must be adapted to changing or newly-defined engineering processes. The demands placed on the incorporation of the framework are largely dependent on how the integration in an existing PLM environment can take place. The vertical integration of third-party products in PLM solutions plays an increasingly important role for customers who progressively expect fully-integrated work processes by PLM providers. Integration should offer a high degree of flexibility. The framework must be able to allow cohesion over all areas of a product life cycle. Therefore, the task of integration is to combine the features of mobile devices with the PLM solution. This requires two conditions that must be fulfilled: Firstly, between the mobile devices and the framework, a uniform data format needs to be defined and used for data exchange. Secondly, the protocol used to exchange data between mobile devices and the framework must be defined. The definition of a unified protocol and a data format is complex, because mobile device providers prefer mostly proprietary solutions, especially for devices which are used in specific areas of responsibility. A device list is compiled in Table B.3 of Appendix B.6.

2.3.5 Expandability and Customizability

The extensibility and adaptability of the framework are further major requirements. In recent years, PLM vendors have attempted to provide a unified software for customers, which is known as OOTB[11]. The approach behind was to offer PLM software products that can be used immediately after installing the software without the necessity of configuration or customization by the customer. Over the years, PLM vendors have recognized a constantly increase in inquiries for customized solutions, which lead to a consideration in their products. PLM customers are distinguished by their high innovation level and want to preserve their individuality. Therefore, the individuality in the PLM software is necessary to support competitive advantage through innovative product development processes with the ability of customization. A high degree of flexibility and adaptability is mandatory to access mobile features by PLM applications. This degree is derived as a requirement for the mobile feature framework.

2.3.6 Reusability

In order to respond to steadily increasing customer needs in a sufficiently flexible way, the question arises of efficiency and effectiveness of the software. Therefore, the reusability of software modules is an important quality characteristic to keep development costs at relatively low levels by reuse and to reduce development times. The reuse of software components can be achieved by the abstraction of modules only without having a dependency to other PLM applications. Therefore, reusable software modules require the definition of standardized interfaces.

[11] Out of the box (OOTB)

2.3.7 Framework

This section describes the requirements for the Mobile Feature Framework which are considered for the reusable design patterns in mobile application. These requirements are specified in the following groups: Fault tolerance, Performance, Scalability, Availability, Maintainability and serviceability, and Reliability.

- **Fault Tolerance:** The framework must be able to respond to unexpected and invalid states. Therefore, incoming mobile events should not lead to a blocking mobile applications or system failures. The stability of the framework must be guaranteed with respect to changes in the mobile application.
- **Performance:** The framework performance is defined in terms of workload, which has to be done within a certain time period. To perform a request, the framework must be able to manage a large amount of events from mobile devices without a dramatic decrease of the reaction speed. Therefore, an effective and efficient framework structure is necessary. Besides, the framework performance also depends on external factors such as the available physical system performance and efficiency of the mobile network.
- **Scalability:** In order to ensure a high level of the scalability, the framework must be able to handle a large amount of events from mobile devices in an acceptable performance. A wide range of individual PLM applications should be able to interact with the Mobile Feature Framework. The limitation of PLM applications and mobile users depends on the scalability of the physical system and the performance characteristics of mobile networks.
- **Availability:** The availability of the Mobile Feature Framework describes the key figure of the reliability, which is used by mobile PLM applications. The level of availability depends on the framework itself as well as third-party systems and (network) resources required by the framework.
- **Maintainability and Serviceability:** Maintainability is defined by IEEE in [IEE90, p. 127] as "[t]he ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment."

Professional frameworks are characterized by the fact that they achieve the operational readiness and maintenance services within reasonable time and cost. The maintainability of the framework must be possible for people who only have a basic knowledge in the subject matter. The modular design of the mobile frameworks and a comprehensive documentation support the staff to perform the maintenance tasks.

- **Reliability:** The reliability of the mobile framework represents a behavioral trait of a constant function that is executed in an intended period of time with the corresponding precision. Therefore, the reliability of individual modules of the framework can depend on the availability of the entire framework. Consequently, there is a close relationship between these two requirements.

2.3.8 PLM Application

The mobile PLM application must rely on the various aspects of mobile frameworks to guarantee the own basic functionality. Thus, there is an important relationship between the mobile PLM applications and the Mobile Feature Framework. Mobile PLM applications have the responsibility for the interaction and communication with mobile users. The framework provides the communication to mobile devices and the access to mobile features.

2.4 Summary

The requirements for the development of a Mobile Feature Framework were considered from three different dimensions. Firstly, the social perspective in Section 2.1 has considered the aspects of privacy, user-specific behavior, and cultural influences in mobile contexts. Secondly, the business requirements in Section 2.2 focus mainly on the needs of different user roles that must be satisfied to accomplish tasks in the PLM environment. Finally, the technical requirements in Section 2.3 were taken into account by addressing the aspects of IT security, connectivity, interoperability with other systems, reusability of components as well as adaptability and extensibility. In summary, it can be stated that requirements are addressed to human, technical, and business needs that cover totally different interests on different levels. For this purpose there is a need to keep the balance between the requirements. Not everything that appears to be technically possible corresponds to the values of businesses and needs of employees. It may also happen that needs of various levels seem to be contradictory and thus a consensus between various interest groups must be found, accepted, and respected.

3 State of the Art

The following chapter considers the applied business research practices for the current state of the art. The author has chosen to focus primarily on performance analysis for the core areas of mobile information and communications technology (MICT) and PLM in context of frameworks that focus on mobile features. New aspects are considered as a result on combining these two areas in Section 3.3 of this chapter. The objective is to examine existing approaches and concepts that are suitable as a basis for identified issues as well as identify existing gaps in the actual state of research. Section 3.1 deals with the field study of mobile communication technologies and mobile users. Section 3.2 presents the context of PLM with existing approaches and technologies. The combination of mobile ICT and PLM is introduced in Section 3.3. Section 3.4 investigates frameworks for mobile applications and ends with a summary evaluation in Section 3.5.

3.1 Mobile Information and Communications Technology

The study of these advanced forms of IT in the particular context of mobile technology is organized into five sections. Section 3.1.1 addresses the subject mobility and contexts with different characteristics. Section 3.1.2 deals with the needs and concerns of mobile users. The surrounding environment of mobile devices, various device types as well as the architecture of mobile devices is discussed in Section 3.1.3. This chapter ends with an investigation of mobile networks in Section 3.1.4 and a summary evaluation of mobile ICT field in Section 3.1.5.

3.1.1 Terminology in Context of Mobility

The terminology for mobile solutions has a large variety and is not always clearly defined. However, the classification of this work into the existing mobile world in a consistent form requires the accomplishment of definition, delimitation, and classification of the mobile terminology.

Mobile

The primary question that arises is to understand what *mobile* actually means. The dictionary describes mobile as the capability of moving. Krannich defines as "the change of the position or the physical or social context"[1] [Kra10, p. 70]. In contrast, *mobility* refers to different areas, that are context specific. In this respect, once a normal user is moving, it can be considered as a *mobile user*. The mobile context has diverse views on environmental, technological, social and, communicative aspects. Mobile devices represent only a piece of the complex ecosystem of mobile users around the world.

Mobile work is characterized by user activities, which are not only exclusively performed in stationary office, but also during or after completion of a change of place. Mobile office describes all elements of the mobile environment, which are required by a person to gain the capacity to work on the move. Mobile work includes tools such as mobile devices, mobile network infrastructure as well as mobile software systems that have been designed for such situations to fulfill tasks. These three components are summarized under the concept of *Mobile Computing*. All tasks which are not carried out on a stationary workplace can be understood as *mobile tasks*. A stationary workplace is characterized by the following features, which provide a prepared installation: a network infrastructure (reliable wired or wireless connection), external AC power supply, furnished premises (with tables and chairs), and stationary computer systems for users. Accordingly, tasks that are performed in such a workplace environment are considered as *stationary*

[1] Original German passage: "Mobil zu sein meint die Position oder den physikalischen oder sozialen Kontext zu wechseln." translated by Jens Michael Hopf

tasks. Tasks that are carried out with a mobile computer system (e.g. laptop) are not considered as mobile tasks in this context, even if the task is performed during a change of place (e.g. in aircraft or while traveling by train), because such systems provide nearly the same user experience as stationary computer systems.

Mobile activities describe the activities of the person within a mobile situation. Thereby, mobile activities can be carried regardless of time, place, activity type, and use of technology. Through a lack of prepared installation as it is the case for stationary work spaces, the execution of mobile activities is a particular challenge for different mobile dimensions (social communication, information delivery, and mobile technologies). Communication of mobile devices takes place mostly over mobile networks such as UMTS and LTE. The access to company information is carried out via Internet depending on the context. The term *mobile Internet* is understood as the provision of the Internet on mobile devices. Mobile devices require a modification of the content for data presentation as well as a reduction of the data load suitable for mobile contexts compare to stationary devices due to other device-specific characteristics (e.g. touch screen, display sizes). Conventional interfaces of stationary computer systems are not suitable for new interaction possibilities between humans and computers in mobile situations. In addition, the challenges of the predictive application concept are considered insufficient and thus will adequately identify the user's needs and adapt the program behavior to the respective user. The interaction between humans and computers in mobile contexts is a research field of mobile HCI.

The context of this work refers to the movement of people who change their position by variations of changes in location as well as constant context changes of specific action patterns. The variation between places and people are related to action contexts. Mobile users can modify the situation not only through changes in place, but also by the movement of other persons involved in their own situations.

Context and Context Awareness

The increasingly intensive interaction with devices in mobile contexts has become indispensable today. Few decades ago, most devices operate stationary while people were forced to adopt the static context of the devices and were context-bound by the devices. In the first glance, mobile users differ not significantly from stationary users. Stationary users are focused on their daily activities while they are staying in the office working with local workstations. Both types of users communicate and access information to consume or contribute contents. Typical examples of stationary devices are workstations and other wired devices such as landline phones and stationary game consoles. The term context is used in various fields and is understood as a key notion, which must have a logic structure to allow context statements about coherences of a corresponding environment [Cf. Lok07, p. 4]. According to the definition used by Duden, the term *Context* is defined as "content-based thoughts related to an utterance and factual as well as situation coherences out of which it must be understood"[2] [Dud13a]. The most commonly used definition of Dey and Abowd defines *context* as

> "[...] any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." [DA00, pp. 3–4]

Context awareness is limited to a specific context, which depends on it [Cf. Dud13b]. This definition for linguistics is quite similar for the software behavior of mobile users. In computer science, the context awareness describes the software behavior that analyzes information about surrounding environments, evaluates, and applies to coordinate the software behavior to individual situations. The context of mobile users is composed of various components. As a result, the actual situation, the current location, the persons involved as well as behavioral

[2] Original German passage: "inhaltlicher Gedanken, Sinnzusammenhang, in dem eine Äußerung steht, und Sach- und Situationszusammenhang, aus dem heraus sie verstanden werden muss" translated by Jens Michael Hopf

patterns play an important role, which cause specific user behaviors. Goodwin and Duranti specify the *mobile context* by the following basic parameters: (1) *settings*, (2) *behavioral environment*, (3) *language as context*, and (4) *extrasituational context* [Cf. DG92, pp. 6–9]. Mobile users are indirectly aware of these parameters, however, it is partially influenced. By the specification of these contextual parameters, it becomes possible to generate new or revise contexts. Modifications to contextual parameters can be performed by the mobile user or by other components that can cause corresponding changes. The presence and behavior in contexts are representative cases. For example, the mobile user takes into consideration when traveling by train, which social (fellow travelers) and spatial (e.g. four-persons compartment) structures are present (settings), which formal and informal rules (e.g. disturbances by a phone) have to be complied (behavioral environment), which language is used for communication (language as context), and if further information (e.g. announcements on the train, surrounding noise) plays a role for the context (extrasituational context). All these parameters represent an own mobile dimension, which is more precisely described in Section 3.1.1. Section 3.1.2 discusses these parameters under cultural aspects.

Mobility Dimensions

According to [Zän00], all three dimensions of mobility are examined from various perspectives. In user's eco-system, the social mobility is generally understood as the exchange of individuals or groups from different socio-structural positions [Cf. MV08, p. 150]. The physical environment describes the spatio-temporal environment, for example, if a person is on foot (short distance and small time period) or starting a long journey (large distance and long period). At the same time, this also describes the spatial freedom to communicate. The informational mobility represents the third dimension, which enables the information exchange without physical movement through appropriate communication media. Table 3.1 shows the mobility forms.

Movement in Time		Movement in Space	
		Yes	**No**
	Yes	Physical Mobility	Social Mobility
	No	Teleportation	Informational Mobility

Table 3.1: Forms of mobility (adapted from [Cf. Tul07, p. 31])

Social Mobility

The main difference of mobile users compared to stationary users is that mobile users are exposed to continuously changing mobile contexts, while external influences affect corresponding contexts which differ especially if they are present in the real or virtual world. Stationary users are also exposed by changing contexts, but they are limited to the local working environment in the office. Ballard calls these contexts as *micro-contexts* [Cf. Bal07, p. 15], whereas Hess describes this subject as *social contexts* [Cf. HFH$^+$05, p. 10]. Cultural and social influences play an equally important role. The codes of conduct are a decisive factor. For example, the preference of a phone call during a current discussion. Therefore, social education plays an important role in the mobile user's attention. How the user deals with the situation, depends on the methods of the mobile user who can respond with his habits, for example: During a meeting, an incoming call is rejected, whereas answering a text message is accepted. Figure 3.1 illustrates the interactions of mobile users with different contexts by using the example of a customer meeting. During a client meeting an employee's mobile device is ringing. The employee notices this incoming call on the mobile device and must decide how to react. He can refuse or accept the incoming communication, or reschedule. Each of his actions makes it necessary to interrupt the physical context. The employee's decision depends on different factors. It matters how many contexts the employee has to manage, which medium has been used to initialize the communication, who plays a certain role in the context, and which cultural conditions the context is applied. In this case, the mobile user needs to manage different contexts and

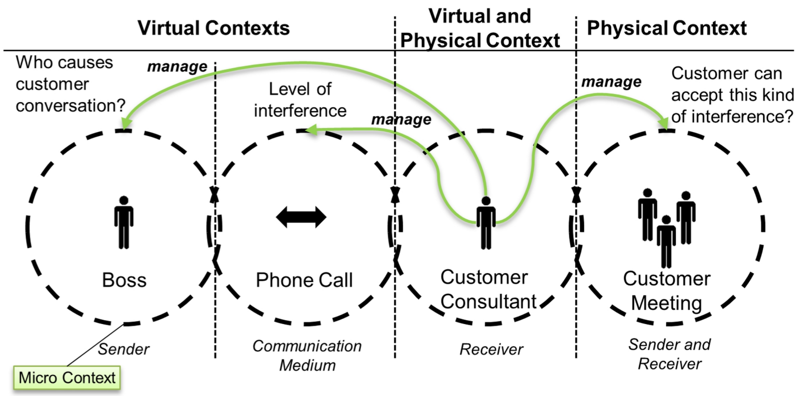


Figure 3.1: Sample: Customer meeting interrupted by a phone call. (source: own representation based on [Cf. Bal07])

has to react to (continuous) changes. Particularly mobile users have a high count of context switches, which could be better harmonized by context management integration in mobile applications to reduce social conflicts between the virtual and physical world. Table 3.2 illustrates the characteristics of mobile users compared to stationary users.

Informational Mobility

As every context has its own characteristics, information must be registered, analyzed, and taken into account in the mobile applications. Beyer and Holtzblatt refer to this kind of information gathering within the context as *contextual inquiries* [Cf. BH98, pp. 41–42]. The informational mobility is concerned with the intangible good "information", which is transported via communication media and consumed by people or objects. The information transmission requires appropriate coding and communication standards. Information cannot however actually be consumed in the proper sense of the word, but the progressive time can age the information so that they lose in importance. Mobile devices could determine the current physical context of the mobile user through mobile features. This information could be sent to an application that can

Characteristic	Mobile User	Stationary User
Disruptions and Contexts		
External disruption has an impact on the user experience (e.g. sun reflection, road noise, low temperatures in the winter season)	Yes	No
External factors can cause context interrupts	High	Low
Complexity of the social context	High	Normal
Simplicity to contribute information	Medium	Low
Consume complex structured information services	No	Yes
Virtual interrupts cause physical disgruntlement	High	Low
User's attention to other people in real contexts	Low	High
Switching between real and virtual context (e.g. real conversation versus phone call)	Yes	Yes
Emotions caused in mobile contexts (e.g. an amusing information in a text message)	High	Low
Virtual context interrupts can cause interruptions of multiple physical contexts (e.g. Person with the same ringtone)	Yes	No
Stress level by interruptions	High	Normal
Count of interruptions	High	Normal
Count of contexts to manage	High	Normal
Count of social conflicts	High	Low
Availability		
Response time of users	Fast	Slow
User feeling more connected to others	Yes	No
User availability	Mostly present	During working hours

Table 3.2: Comparison between mobile users and stationary users (adapted from [Cf. Bal07] and [Cf. Gid97, pp. 119–121])

administrate the disturbances. It would also be conceivable to combine the physical and virtual contexts. For example, new information could be synchronized in a virtual and physical context. Therefore, respective communication media is necessary to ensure the data exchange.

Physical Mobility

Physical Mobility (also called spatio-temporal mobility) represents the location change of objects within a certain time period and depends on the social and information mobility. Objects can be, for example, persons (among other mobile users) or things (among other mobile devices). Thereby, the physical presence is no longer required for the communication and thus the *physical absence* represents one of the cornerstones of the remote society [Cf. Hel07, p. 70]. A subset of the physical mobility includes the *personal mobility*, *service mobility*, *session mobility* as well as device mobility which are closely linked as illustrated in Figure 3.2.

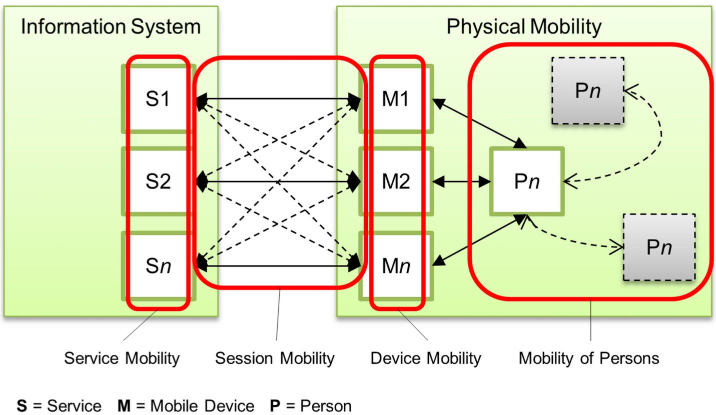


Figure 3.2: Mobility dimensions

The *Personal Mobility* describes the possibility to identity people through authentication with a mobile device against stationary information systems to consume appropriate services. The *Service Mobility* allows mobile users to consume any service from a stationary information system, regardless of the current location, the person, and mobile device used. The adaptation of provided services depends on the mobile network and the mobile device.

The *Session Mobility* is defined as the ability to enter a temporary relationship between the mobile device and the information system. The session is required to consume a provided service of the information system to fulfill appropriate (distributed) tasks. The session can have different states (e.g. created, paused, resumed, and terminated), which are influenced by a number of factors (e.g. a person switches to another mobile device but continue to use the service; the network link to the service is interrupted; the session is terminated due to time constraints).

The *Device Mobility* is characterized by the ability of the spatial variation and mobility. Instead of the device mobility, the term portability is also used [Cf. Sch03, p. 15]. Bulander has splitted the group of device mobility into *serial mobile* and *mobile* [Cf. Bul08, pp. 11–12]. In this work, the terms *generally mobile* and *qualified mobile* are used for better understanding. *Qualified mobile* describes devices that have been developed and optimized for mobile use without an external power supply and can maintained a permanent communication link over mobile networks. These types of devices are situated in a kind of standby mode and are immediately usable for users. For example, this applies for mobile phones and tablet PCs. In contrast, *generally mobile* devices are turned off and need a start-up time to reach operational status. These devices are also mobile, but are mainly used at stationary locations. In this case, the mobility refers to movement itself or the transport that performs the location change [Cf. Bul08, pp. 11–12]. A mobile user has a richer view than a stationary user with the consequence that mobile users are changeable and unpredictable as stationary users. External influences can change human behavior in relation to the current context, which causes a much higher complexity in dealing and administration of user contexts. Thus, the correct interpretation of the user context is complicated because important information is not always accessible and visible. Future technical and organizational approaches must have the ability to analyze mobile contexts by appropriate methods to provide this information available to other applications.

3.1.2 Mobile Users

This section examines the aspects of privacy and cultural characteristics. Beyond this, the section will consider the individual behaviors of mobile users such as privacy and culture.

Privacy of mobile Users

Additional mobile features such as navigation, sensors, digital camera, and media playback provide a series of new opportunities for mobile applications to users. Mobile users were now in a position to choose between text-based communication techniques such as SMS, MMS, e-mail, chat as well as media-supported communication technologies, including internet telephony (VoIP) and video conferencing used in apps. Thereby, external interfaces such as WLAN, Bluetooth, USB, and Infrared ensure the communication with the outside world. The integration of mobile features directly into the mobile application allows more accurate information collection and sharing for the mobile user who in turn leads to an improved user experience. New opportunities in applications are achieved only by including collected data from mobile features and the subsequent transfer and processing of personalized information. Only by personalizing[3], it became possible to customize the user interface of apps and to deliver pre-selected content to the mobile user. This resulted in a simplification in the operation and use of information in a personalized mobile environment. The improved user experience and the more intensive use of smartphones led to an increasingly personalized data collection and analysis by app providers. Nowadays, apps can access the address book of the user, perform site localization, and record

[3] Personalization is means to customize content and services to meet the needs of the user. Collecting user information is defined as a process in which the data resulting from the interaction with the user. The user should be assisted in accessing, retrieving, and storing information as well as managing personal preferences and characteristics.[Cf. SX07, p. 440]

user activities. By a low security awareness of mobile users, application providers are able to collect more personalized information. Companies have also expressed concerns over the possibility of data collection so that nearly 57 percent of the companies consider privacy as a particular challenge[4] [Cf. Chr12, p. 127].

About 76 percent of respondents in a survey in 2013 about safety of apps believe that apps provided by app platforms are safe [Cf. MKG13, p. 50]. Users tend to ignore repeated warnings when performing a task [Cf. MHB10] and blindly accept warnings similar to EULA[5] messages. Once mobile users have the feeling that the mobile communication is secure, many tend to be less careful with security measures [Cf. AK12, p. 218]. In contrast, users who have a low sense of security avoid the storage of personalized data on the mobile device [Cf. AK11, p. 636]. Due to a versatile use of personalized data, there was a series of data privacy protection scandals (see [CB12, Vel12, Gol11]). As a result, the attention of mobile users dealing with this issue has been increased. Today the topic is met with a greater interest. By a transparency lack of app providers, mobile users have limited knowledge about whether and which personalized data will be transmitted and processed by providers. Before a mobile application installation starts, a notice to the mobile user about data recording is made but will not be discussed in detail in what form and intensity the data processing takes place. In addition, the majority of apps do not allow the mobile user to influence the data collection and transmission. As a result, the acceptance of mobile users decreases when personalized data is transmitted to app providers, even if an improved user experience can be achieved.

In a survey by the Pew Research Center, they found that 54 percent of users would not install a mobile application if it were known how much personal information has to be shared to use these applications. 30 percent of users said they have deleted an already installed mobile application from their mobile phone, because they learned that collected personal information should not be shared with others [Cf. BSM12, p. 2]. Therefore,

4 The empirical survey of IT decision makers (mainly CIOs) was conducted in 2012 among 800 companies.

5 End-user license agreement (EULA)

app providers are required to communicate detailed information about the form and scope of the submission and processing of personalized information and deliver advanced options that prevent personalized data transmission. App vendors are mostly reticent about additional privacy options because this has a negative effect on its business model especially for advertising funded apps. The trade of personalized data collected through mobile business applications is not the main interest but rather a demand to fulfill the mobile productivity in companies and to improve the mobile user experience. It is therefore essential to communicate clearly what time period information is collected and for what purposes data are processed. The obtained benefit needs to be communicated clearly to the mobile user, which can be reached only by temporarily capturing of his personalized information. If the user wants to consciously decide against the benefits, the app must provide an option to switch off the collection and transfer of personalized information and have to offer an alternative function to the user. Following scenarios are examples of using the benefits of collected personalized data and a corresponding alternative solution: The phone camera could be used to scan a barcode, whereas the alternative solution would offer the direct input of the barcode number. The search for location-related information may be done by detecting the temporary location of the smartphone, while an alternative solution would provide the manual input of the user location or an unfiltered data search.

In general, alternative solutions would mostly reduce the user′s mobile productivity. Therefore, the goal should be to act against these concerns and fears of mobile users by providing them with additional information to increase the user acceptance. Only essential personalized information should be recorded for the implementation of the currently executing function. In addition, the mobile user must be given the opportunity to delete already captured personalized data. The user acceptance depends decisively on the user′s motivation for the use of the mobile application with its mobile features.

Mobile User Culture

While generic terms such as internet, computers, and TV are understood worldwide as the same, this is not the case for the term *hand-held phones*. Depending on the region, the term has a different meaning and understanding. Germans and Finns like to use the term *Handy* and *kännykät* because the compact and lightweight design fits perfectly [Cf. The09] in the hand, without the need to place the device on a table in order to use it. In this case, the external characteristic is used to define the term rather than the technical aspect of the device. In the UK, the device is called *a mobile*, whereas in the United States it is a *cell phone*, in Latin America *celular*, in Bangladesh a *muthophone* (phone in the palm of your hand), in Sweden *nalle* (teddy bear) and in Israel a *Pelephone* (wonder phone) [Cf. Can12]. In turn the Chinese defines the device as *shou-ji* (hand machine), while implying a technical machine with a set of specific functions. In Japan, the term *keitai denwa* is used (also known as sub-term *keitai* for portable), which refers to the portability of the device [Cf. The09]. Since enormous cultural differences can be found in the term definition of mobile phones, more variation is observed in the mobile users behaviors. Currently there are different opinions regarding personal communication technologies of mobile users. James Katz of Rutgers University in New Jersey is convinced that people prefer standardized infrastructure and gravitate to common preferences with universal functions. People from different countries show mostly the same behavior with only small national differences. This view is also shared by Nokia, whilst mobile users are investigated in categories instead of geography. However, this view does not apply for mobile content. Kathryn Archibald from Nokia, said: "We see more differences within countries than between them. [Also] [w]e need to operate globally, but be relevant locally" [Cf. The09]. In Section 3.1.2 of this work, the individual preferences of mobile users are investigated. Because specific parameters have an influence on the user behavior and thus the type and manner of use the user behavior is investigated in Appendix A based on the dimensions: Political conditions, Financial and economic conditions, Formal education, Generation, Social relationships, Health aspects and Physical environment.

Individual Mobile User Behaviors

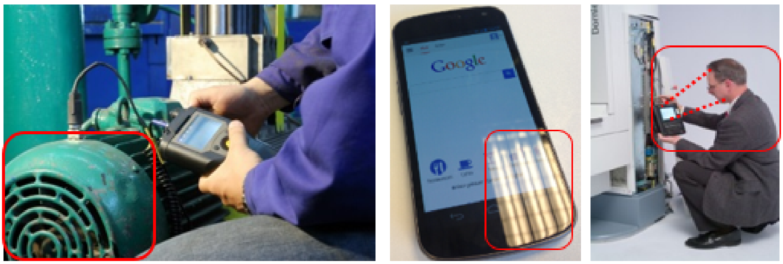
This section examines the individual behavior of a single mobile user. Therefore, situational contexts, the nature of a mobile user, and the use of mobile phones are analyzed as well as different context switches such as situational changes and device changes (e.g. from a smartphone to a tablet pc) are treated. The research field of mobility has been investigated by the two perspectives of the macro-level and micro-level. From the macro-level perspective, the mobile users are analyzed in the context of spatial mobility. The spatial mobility is understood as the movement of people, for example, when users move around using public transportation to make a location change. The perspective on the micro-level deals with the available resources, the cultural and social aspects as well as the user's individual characteristics.

Gestic

The input interfaces of mobile devices have developed rapidly in recent years. During the turn of the millennium, the interaction with the mobile phone was carried out using keypad and the use of function keys and pens for the operation with pocket PCs was followed in the years onwards. Pocket PC which is based on Windows CE operating system (WinCE) has been developed for embedded devices and provides an appropriate surface with pre-installed applications. Because Windows CE was designed as a real-time operating system, the focus has been originally set for industrial applications, although the perception of the public is on the mobile phone. Windows Mobile, which is based on Windows CE, was introduced in 2002 and had no multi-touch support. A pin and corresponding keys were needed to interact with the display. Today's smartphones have touchscreens as input interfaces that allow control by hand gestures. Future smartphones will allow non-contact gesture control for example, incoming calls must not be accepted by pressing the "accept call"-button anymore, but the call will be accepted automatically by holding the device to the ear. In the industrial sector such gesture technology is far less common. The user input is realized either via function keys or touch screen. The operations are structured via menus and items. Fulton [VWW11] describes in a reference card various gesture types that are analyzed and evaluated in Appendix B.4.

External noise

Other situations and environmentally conditional influences are external noise, intense light sources, and spatial conditions. Intensive external noises may cause that information (e.g. audio messages, video clips, phone calls) is only partially perceived and thus the risk of misinterpretation by incorrectly transmitted information increases (see Figure 3.3a). A similar information loss occurs when intense light sources affect the communication of visual information (see Figure 3.3b). Thereby, information cannot be read correctly and incomplete information may lead to wrong decisions. Spatial conditions are understood as the spatial effect by the external environment of the mobile user. A lack of space and limited space have an influence on the body posture and position of the mobile user and thus having influence on the interactions with the mobile device and the visual perception of information (see Figure 3.3c).



a) *Noises* **b)** *Light reflection* **c)** *Spatial condition*

Figure 3.3: Situational and environmental influences. a) adapted from [Cf. Edw09] and c) adapted from [Cf. RFI13]

Type of interactions

Smartphones expand the ability to constantly stay connected and communicate as needed. In contrast, mobile phones only provide basic functionalities such as phone calls and SMS whereas smartphones provide more complex tasks such as process monitoring and process control. Some studies (see [SRKH06, MPT09]) have already shown that users use their mobile devices for different purposes. Users tend to adapt their mobile devices to their personal needs, as far as the technical possibility exists. For example, modification of the font type and font size, customization of the screen background and sounds for events such as incoming e-mail or battery level as well as the arrangement of applications on the screen for a particular user scheme. The needs of mobile users are similar to those of desktop users, because desktop users make adjustments on a comparable way, even though they are not in the same intensity as mobile users who are conditioned by the higher density of personalized information. The biggest open African social network Mxit considered the home screen of the smartphone owners as follows:

> "A splash screen [on a mobile device] can be compared to the cover of a magazine. It is the first thing a user sees when logging in [...], even before getting into a chat with friends." [Mxi12, p. 4]

In this regard, the social network Facebook has provided, in April 2013, software for various mobile operating systems that present relevant information of the social network directly on the lock screen instead of home screen. Such approach is currently not available for PLM information on the market, even though this would allow a rapid access to context-sensitive information in a mobile environment. In addition to the personalization, mobile devices are also used to carry out everyday tasks. Upcoming tasks are not always accomplished on a single mobile device. Nowadays, it is common to switch between multiple mobile devices having different form factor to consume information and to do limited contribution. In this manner, the smartphone enables users to perform tasks that would otherwise have been done later on a full-featured workstation [Cf. MPT09, p. 4]. In particular, social networks, microblogging services and, cloud services have already realized such multiple access scenarios. For such applications, it does not matter which type of device is being used as a medium for interaction. For example, Dropbox[6] allows a seamless exchange of documents and files between all mobile and stationary devices. A much more complex challenge is the continuation of tasks that have been started on one device and continues on another mobile or stationary device. Once a task is started on a workstation, it cannot always be continued seamlessly thereby, personalized and runtime information are often missing which are mandatory to reopen documents and user views as well as recover communication links. A lack of interoperability, openness as well as security aspects additionally complicate the realization of such scenarios. For example, an initiated chat communication cannot be seamlessly moved from one workstation to a mobile device and must be reinitialized. Therefore, an interruption of the already active communication connections on the workstation is required. In addition, the user may be faced with a different user interface and reduced functionality, which in turn can affect the user experience. Other examples have been mentioned in a study by participants who have created a draft e-mail on the computer, synced the composed e-mail to the mobile device in order to complete and send it later [Cf. MPT09, p. 5]. The need for such scenarios have been already identified in previous researches (see [DP08, POS^{+}01]), but smartphones have not yet received sufficient attention in this field, so that further research appears necessary in this area. Thus it becomes necessary that personalized data, such as customized settings, data views, and links between communications are synchronized to compatible device types to ensure the ease of use. A lack of synchronization of such data would lead that users are limited in performing tasks. In worst case scenario, the user would have to manually transfer data from one device to the other device or data would have to be recreated, the communication links have to be rebuilt as well as the data views have to be organized again and thus a huge part of the mobile user's productivity would be lost.

[6] Dropbox is a proprietary cloud service that was established in 2007 which allow users the synchronization of files and folders between different operating systems.

Individual usage

Mobile devices are mainly used by users to consume information and to coordinate activities [Cf. SRKH06, p. 413] [Cf. MPT09, p. 5]. The contribution usually takes place in a *time-saving interactive form*. This contribution form can be for example, the creation and sending of pictures, videos, and graphics. Another example is described in relation to the device authentication in Appendix B.3.

A *time-consuming and interactive-less form* of contributions are considered as reviewing and processing of documents, which is less frequently performed on mobile devices, even if the device has already the technical capabilities. The interaction and control are primarily carried out through selection menu, buttons, and icons. In this category, the mobile search may also be classified because the manual input of search terms is required by the mobile user. An empirical analysis of mobile search queries based on the data of a search provider indicates that more than half of mobile search sessions contain only a single search and thus does not include mobile search to an everyday activities of a mobile user [Cf. Voj08]. By a missing full size keyboard (Soft or on-screen keyboards normally used) the input of information is less comfortable as it is the case for workstations. Therefore, the most mobile users tend to contribute with short answers only when it is required, while more detailed answers are formulated on the workstation [Cf. MPT09, p. 8].

Several studies have been investigated including tasks that are carried out on mobile devices for obtaining more knowledge about mobile users performing productive tasks on mobile devices. In a study by Karlson et al. [KMJ+09], the users have shown a different behavior in the use of applications between mobile devices and workstations. By logging the activities it was observed that users often used the appropriate applications for corporate e-mail and Internet access on both devices. These results are consistent with another study of [MPT09]. Device-specific functions such as SMS and phone calls were mainly used on the mobile device, while the file manager, instant messaging and office applications have been preferable used on stationary devices. The results of this study can possibly be explained by the fact that not all desktop applications are available as a mobile application and the contribution does not offer the same user experience as it is on stationary devices. In the study it was also noted that participants access business e-mails outside of normal working hours[7] and according to Karlson et al. it is inferentially be seen as part of the daily routine. All interviewed participants reported that they carry their mobile devices with them [Cf. MPT09] in order to take advantage of the mobile phone to access business e-mails. Some participants argued in this regard as follows:

> "If I am headed to work, then I usually want to know if I am headed into any problems. [..], if I can resolve [a problem] on my phone, then I just do it. It is more convenient to go ahead and do it, and then I can forget about it. If I can answer it quickly and I have the time, I might just punch it out." [KMJ+09, p. 405]

Based on their findings, it can be inferred that participants would enter other information systems to access corporate e-mails when appropriate technical possibilities exist. Due to the portability of mobile devices, a space for a broad independent usage is created to increase the motivation of mobile users to use the device. Thereby, access to information systems must be carried out quickly and efficiently in order to exploit the short and limited time windows of mobile users (e.g. while traveling [Cf. MPT09]) as productive as possible. Presently, users perceive the smartphone more as an information medium than a replacement for the classic stationary workstation [Cf. MPT09, p. 5]. Therefore, both types of devices should know their respective strengths and weaknesses and act within the meaning of the user. Currently, a lack of suitable methods and concepts exist that allow a seamless and interoperable integration for existing information systems to integrate mobile devices more deeply into work processes and tasks.

[7] Over the period of about 226 days was observed that all participants who have used both devices in the same day, 88.1 percent of the participants preferred to use the phone before the desktop. In addition, the mobile phone was accessed in 87.6 percent of these days after the desktop PC was used. [Cf. KMJ+09, p. 403]

Previous studies have already examined the change of communication media (see [Ret03, DHN02]). While pagers were used in the early 1990s, the communication shifted by increasing popularity of mobile phones to the SMS. A similar change of the communication channel is currently perceived by SMS to mobile data based service like instant messaging applications, which are mentioned in a published study by the market research company Informa Telecoms & Media [CD13]. The use of different communication media represents only one component of the mobile task management. Mobile working means at the same time that tasks must be practicably on several devices based on situational context. Therefore, the approach how tasks are performed by users differs significantly. While part of the users prefers to organize paper documents, others prefer a digitally organized structure of documents [Cf. Mal83]. Another user behavior shows the use of multiple monitors.

In a study by Grudin [Gru01] it has been found that users use their displays for distributed tasks. While the primary display was used for documents, the second display has been widely used for the communication with other people and events by users. In order to understand how mobile tasks are being performed it is necessary to develop the following dimensions: The associated space, the user profiles, the task to be accomplished, and the used device type. These dimensions are represented by corresponding elements (see Figure 3.4) and are dependent on each other. This means that the applied role (user profiles) defines the mobile tasks. In turn, the mobile tasks depending on spatial dimension in which they are carried out. Accordingly the space and the associated connectivity have an influence on the used device. In addition, soft factors also influence the user behavior, so that they prefer to perform a task on a pre-selected (mobile) device. In a study by Oulasvirta and Sumari [OS07] the behavior of participants was observed on how they use the device and how often they switch between devices over the day to perform tasks. In case that a task was not tailored to the screen size, a change was preferred to a dedicated device with a larger screen size (e.g. newspaper). A device switch was also triggered from the spatial dimension for reasons of privacy.

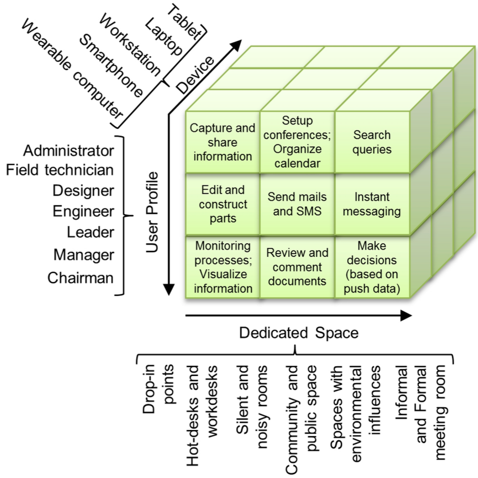


Figure 3.4: Mobile task dimensions

Time aspect while interact

The time plays a further important part which is essential to cope with tasks. Smartphones are particularly well suited to perform less-extensive tasks which can be completed within few minutes. In this context, information workers spend an average of 3 minutes to process an assignment [Cf. GM04, p. 113]. Quickly performed tasks can be, for example, checking e-mails, querying the status of a process or the note during a meeting[8]. Devices that are suitable to cope with more comprehensive tasks require more time to reach the operational readiness. The operational readiness is defined by the device preparation with the

[8] In a study by Oulasvirta [OTRK05] it was investigated that mobile users switch their attention between mobile devices and the surrounding environment in dependence to the assigned space for the fulfillment of tasks. During the interaction with the mobile device, the participants had to change their speed of movement in order to fulfill mobile tasks [Cf. OTRK05, p. 925]. To compensate the lack of attention for the surrounding environment of mobile users, systems such as CrashAlert by Hincapié-Ramos and Irani [IHR13] has been presented which attempt to notice the user while walking. Nevertheless, these systems are unsuitable for practical applications through a bulky hardware setup and the native detection of obstacles.

entire associated peripherals device and commissioning. The decision of the user on which type of device is used in the current context, depends significantly on the available time interval. Appendix B.5 presents an example of activity profiles for a mobile user versus a stationary user. In summary, the properties of actual user behavior listed in Table 3.3 can be identified as trigger for switching devices as well as reasons for multiple devices usage. The consideration of individual behavior patterns of mobile users represents an enormous challenge. In order to achieve an improvement of the user experience in mobile situations, the relevant dimensions to carry out mobile tasks must be considered sufficient. In this respect, Section 3.1.3 analyzes mobile devices and how this is considered by the surrounding user environment and the user behavior while performing mobile tasks.

Categories	Properties
Device properties (Dimension: Device)	- Screen size - Connectivity - Physical form factor - Performance and Security
Personal ergonomics (Dimension: User Profile)	- Preparation time of the device - User experience to fulfill tasks - Personal acceptance of devices - Ability to multitask - Fallback device and data backup device
Social interaction and location (Dimension: Dedicated Space)	- Device acceptance in certain contexts - Privacy to perform a task - Surrounding noise

Table 3.3: User behavior properties that trigger device switches

3.1.3 Mobile Devices

Mobile phones have an eventful history, Figure 3.5 illustrates the different stages of innovations in mobile technology. This figure shows that the count of technology innovations has been increased in the last years. This applies in particular for the transformation of hardware components into software logic, mobile communication standards, and also for changes in the user interface to touchscreen. Since the enormous economic potential of Short Message Service has not been recognized in time [Cf. HL05, pp. 76–77][Cf. HHHT10, pp. 131–132][Cf. Bal07, p. 22], Wireless Application Protocol (WAP) could only partly succeed [Cf. Dor05, pp. 1211–1212]. The success of touch screens was initially also very limited. The worldwide success of the iPhone from Apple in 2007 brought the touch screen technology on smartphones the breakthrough in the mass market. Nowadays, mobile phones are mutated into multi-function devices, which not only cover the pure communication, but also allow complex tasks and interactions. Mobile users are adjusted by their own historical experience skeptical about mobile websites. One reason for this was that the user experiences in WAP websites compared to desktop computers were too different [Cf. Bal07]. Small screens of mobile devices, data entry capabilities, and a limited bandwidth influenced the user experiences negatively [Cf. TP01, p. 1122].

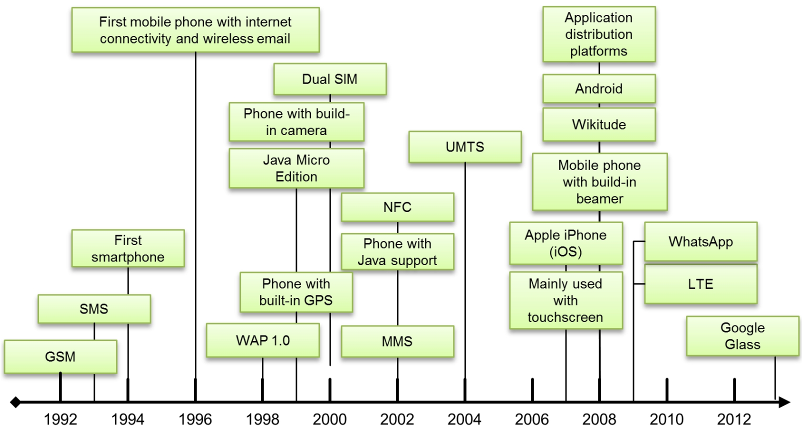


Figure 3.5: Mobile technology innovations (source: own representation based on [Cf. Hub02, p. 20] [Cf. Gra97, p. 197] [Cf. Duf13, p. 6] [Cf. Ste02, p. 367] [Cf. MS00, p. 15] [Cf. Luc08, p. 13] [Cf. Mar00] [Cf. Höl04, p. 102] [Cf. Kol11, p. 18] [Cf. LR10, p. 4] [Cf. Jen08] [Cf. Chu08] [Cf. Mad11, p. 26] [Cf. Bar08] [Cf. Wha09] [Cf. HT12, p. xv] [Cf. Sha12] [Cf. Kac11, p. 193] [Cf. Gog06, p. 144])

Definition of the Term

The term mobile device has a wide scope and is perceived by the public mainly as a smartphone. Thereby, a mobile device can represent a laptop, a smartphone, a MP3 player, a measuring instrument as well as other portable devices. Mobile devices differ considerably from the intended application. Some of these devices have focused on a single purpose, while others represent multi-function devices. The definition at what point a device is mobile or not depends on a variety of partially soft factors. Such factors are specified whether the device is portable (size) and can operate without an external power supply. It does not matter if the mobile device interacts with users or can operate independently. Mobile devices typically have several I/O extensions. For example, an external keyboard, mouse, memory cards can be attached to the laptop, whereas an external headset, GPS receiver as well as measuring instruments through external measuring sensors and measuring boards can be extended to the smartphone.

Types and Classification of Mobile Devices

Many mobile devices are spread across all industries and have found their particular use. Figure 3.6 shows the most common classification. Some mobile devices are used not only for the execution of business processes but can be used for the detection, measurement, and verification of data. Moreover, some of them save human lives. The variety of mobile devices seems to be endless. Since the majority of mobile devices are mobile phones or smart phones, particularly this device group has a large number of personalization. As a result of personalization, there is a closer relationship between the mobile users and the mobile devices.

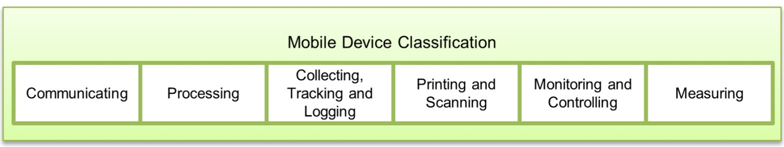


Figure 3.6: Mobile device classification by specific purpose

Such personalization depends on the respective mobile user with his related cultural background, purpose of the device, and the technical degree of customizability of the mobile device. Table 3.4 illustrates the most common characteristics of mobile devices compared to stationary devices. In Table B.3 of Appendix B.6 lists the different classifications for mobile devices and their respective applications with a short description. The majority of mobile devices are diverse and partially optimized for the specific application. Moreover, many devices are proprietary and thus represent a major challenge for companies in the integration into business processes without restrictions.

In general, mobile device types can be subdivide into device ergonomics, device characteristics, and the place of interaction (see Section 3.1.2). While smartphones are held in the user's hand, wearable devices are worn with or on top of user's clothing. Wearable devices operate decently and perform tasks in the background. Thus, this device class integrates seamlessly into the activities of a user without interrupting already started activities. For example, a smartwatch can present information about incoming calls discreetly during a meeting without the need to take out the phone from the pocket and to switch on the device to determine if the information received was important or not. In addition to smartwatches, there are a number of other wearable devices which have to fight more or less with usual technical teething problems of every brand new technology. For example, Mann describes for the data glasses Google Glass that the superposition of live images is problematic, because of the perspective between the eye and the camera [Cf. MALW13]. In addition, the eyeglass wearers would be forced to focus on it with one eye between the distances while the other eye focuses the display with a fixed distance. These kinds of problems do not have smartwatches, therefore the battery life represents an enormous technical challenge because of the small device dimensions for batteries. As already observed in smartphones, the individual components such as touch panel and chips for wireless radio links represent the largest electricity consumers for smartwatches. Although there are already technologies for radio links such as *Bluetooth 4.0 Low Energy* but the required software drivers are not supported or provided by all mobile platforms. The result is that older data transmission standards must remain activated in parallel, which in

Characteristic	Mobile Device	Stationary Device
Device Environment		
Independent device usage	Yes	No
Predictable environment	No	Yes
Complexity of the context	High	Low
Device interaction with other people (e.g. display flip)	Yes	No
Data management	Requires to save immediately	Saves temporary before commit any changes
Attention on application design	Touch or button optimized	Mouse optimized
Server infrastructure as backbone	Yes	Yes
Share data between applications	Difficult	Yes
Degree of personalization	High	Normal
Device Characteristics		
Display sizes	< Medium	Large
Full-sized keyboard	No	Yes
Battery	Yes	No
Integrated hardware features	Yes	No
Performance (CPU)	Medium	High-end
Portable	Yes	No
Impact on application arch.	Yes	No
Hardware platform variation	High	Limited
Variety of information on the device screen	Low	High
Device Availability		
Connection drops	Yes	No
Connection speed	Medium	High
Device availability	Always Online	Working hours
Push information to device	Yes	Yes

Table 3.4: Comparison between mobile devices and stationary devices, based on datasets by [Cf. Bal07]

turn shortens the battery life of the devices. In addition, the short distance for data transmissions proves especially in confined areas as problematic when smartwatches cannot operate independently from a smartphone and thus the connection must be regularly interrupted and rebuilt. A dedicated and standardized coupling mode between smartwatches and smartphones would reduce this problem, but currently there are no efforts in this regard by manufacturers. Section 3.1.3 describes the system architecture of mobile devices and their mobile features.

Mobile Platform Architecture

Mobile platforms are characterized by a uniform base on which applications can be developed and executed. The mobile platform is situated as component between the underlying hardware and the overlying applications and provides a number of interfaces for the interaction. The goal of mobile platform concepts is a logical abstraction between hardware-specific and application-specific aspects. However, in practice it can be recognized clearly that mobile platforms like iOS and Android cannot operate completely independent from hardware and software components and thus the need for a multi-vendor and multi-platform system solution cannot be fulfilled sufficiently. Mobile devices are generically divided into four layers which are illustrated by Figure 3.7.

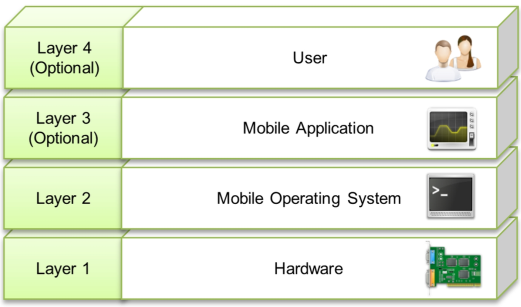


Figure 3.7: Mobile device layers

- **Layer 1:** The hardware layer is the base for all higher layers and represents the most basic functions. Thereby the mobile user interacts with some components of this layer directly. For example, components, such as display, touch screen and hardware buttons. Other components have received less attention although they have been known in this layer or have simply been neglected, such as the antenna or the battery. Another group which is completely transparent to the mobile user includes the processor, memory, or graphic controller chip. However, all of these components play an important role for the user experiences.
- **Layer 2:** The mobile operating system represents the layer between the hardware and the overlying applications. The interactions between mobile users and the applications are coordinated with the hardware. The mobile operating system represents a set of basic functions such as background processes, notifications, storage support, messaging support, sensor support, and other functionalities. The type and amount of basic functions depend on the mobile device and its purpose of use.
- **Layer 3 (Optional):** This layer mainly addresses applications or extensions which are added to the second layer. Thereby these applications primarily use the mobile OS to get access to hardware functionalities. This layer is identified as optional, because not all mobile devices provide this layer (e.g. a conventional pager does not provide the capability to integrate extensions since such devices were designed for a specific purpose of use).
- **Layer 4 (Optional):** The user-layer is mainly known as the user himself whose actions and external influences are dependent on the specific context. For example, the behavior of the mobile user depends whether a mobile device is exposed to strong light or cold temperature. In the first case the mobile user cannot see all elements on the display because he has a limited view and in the second case the user could use gloves that limit the user input. This layer is particularly vulnerable to errors since the unpredictability of human action might probably have a high error potential and is optional because not all mobile devices require an interaction with the mobile user (e.g. sensor stations which have the primary task to transfer measurement data without the need to interact with users).

Mobile Features

Mobile features represent capabilities and services of a mobile device in order to generate added value for the user. Conventional mobile phones offer only basic functionalities like phone to each-other and save contacts on a SIM card, while feature phones provide advanced capabilities such as camera and web browser. However, feature phones do not provide the width range of capabilities as smartphones. Smartphones are considered high-end devices because the higher purchase price compared to low-end feature phones warrant additional hardware feature and thus better hardware equipment. In addition, feature phones are mainly provided with a restricted manufacturer specific firmware. However, smartphones provide a mobile platform with an extensive API that allows the provision of own mobile applications. In addition, the API integration to access hardware-specific features of the mobile platform is more efficient. Initially, a clear distinction and definition for the terms hardware feature (HF), software feature (SF) and Mobile Feature (MF) are carried out. A hardware feature represents a hardware-specific technical capability which is not supplied by the mobile platform, but managed and controlled. For example, a camera consisting of a lens, image sensor and other components. A software feature is an ability that is completely provided by logical software-based components. This type of capability does not require hardware specific characteristics, such as a pocket calculator or a file manager. At work, a mobile feature is seen as an abstract element, which is derived and composed from hardware-specific and software-based capabilities. However, it is not necessarily required for a mobile feature to be derived from both types of capability. Table 3.5 illustrates this aspect.

Example	Mobile Feature = Software Feature + Hardware Feature
1	Object Identification = Object Recognition + Camera
2	Persons Localization = Personal Identification + GPS
3	Language Assistant = Speech Recognition + Microphone
4	Authenticator = Fingerprint Recognition + Scanner

Table 3.5: Mobile feature examples

3.1.4 Mobile Networks

Mobile networks are classified into four categories depending on the transmission range (see Table B.2 in Appendix B.2). Transmission techniques are used according to the application purpose with regard to factors such as bandwidth, reliability, and response time depending upon the characteristics.

- **Wireless Personal Areas Networks (WPANs)** are used for mobile devices for the transmission of data over short distances, wherein the transmitter station is located in the immediate vicinity. The low power consumption, which are a special characteristics in the mobile sector, are used in the construction of ad hoc networks to connect multiple devices. In this case, classic scenarios are data exchanges between mobile devices as well as networking of external devices such as headsets and GPS mice.
- **Wireless Local Area Networks (WLANs)** are used in the local space for the construction of infrastructure and ad-hoc networks. Mobile devices prefer to use WLANs stationary facilities (e.g. in the office) to load a larger amount of data. While infrastructure networks require as entry point a wireless access point (WAP), ad-hoc networks perform the cross-linking of devices directly.
- **Wireless Metropolitan Area Networks (WMANs)** are extensive regional networks, which offer access through various access points using the so-called hot spots. This technology is interesting for devices in mobile situations when a large amount of data must be loaded. WMANs are mainly deployed in densely populated locations (e.g. cities, airports, train stations) which have a comparable bandwidth to wireless networks. However, the network must be shared with other participants.
- **Wireless Wide Area Networks (WWAN)**: Mobile devices mainly create communication links to mobile networks of the WWAN category. Therefore, the spatial context plays a minor role, because the network coverage of the mobile networks already exists over a larger geographic distance. However, the bandwidth and response times vary significantly depending on the mobile network and other technical characteristics compared to other networks.

3.1.5 Summary of MICT

The mobile sector consists of the partial areas of mobile users, mobile device classes, mobile platforms, mobile networks, and mobile applications, wherein each area includes their own complex structures and challenges. While the main focus of mobile users is on specific behaviors and cultures, mobile device classes focus on technical hardware features and mobile platforms to the manner in which they provide capabilities to support mobile devices and users through appropriate functions. Mobile networks play a key role when users want to communicate with mobile devices. In this respect, criteria such as the efficiency of the mobile network are crucial. Finally, mobile application software is the crucial component, because though the combination of all other components it enables the integration to other systems and allows collaboration with other participants. To achieve this objective, mobile application software must perform the interactions with mobile users and the mobile platform as well as create and manage communication links to other system instances through mobile networks.

In summary it can be said that the context of mobile users is considered inadequate in mobile applications. Typical user behaviors are only considered in a small scale by manually configurable elements, while cultural backgrounds play almost no role in mobile applications. The use of mobile features is a manageable framework and can be found mainly in the consumer sector, while industrial application scenarios favor predominantly conventional methods such as manual text entries and selection menus. In this case, there is a considerable need for research, especially for industrial applications. The fragmentation of mobile platforms and mobile device classes are an increasing complex challenge in terms of a homogeneous mobile system environment. Therefore, new mobile platforms and device classes require flexible structures of the entire system solution to realize integrations with a minimum of time and expense. Mobile networks have undergone continuous development over the past years while increasing the efficiency, but the demands on required bandwidth, response times, and network availability increased at the same time. The introduction and expansion of new mobile high-speed networks is mandatory to meet the increasing demands especially under the aspect of the use of mobile features.

3.2 Product Lifecycle Management (PLM)

The concept of product lifecycle management typically involves the entire company. The management of a large number of engineering tasks requires the implementation of a strategic concept, which integrates and coordinates all persons involved, processes, procedures and resources. PLM has many points of view and thus it is almost impossible to make a simple introduction through the IT department of the company without involving the departments into the product development cycle.

The PLM concept starts with the initial idea and continues with design and engineering, construction, service as well as marketing, distribution, and disposal. A distinction is made between the functional view of designers, the production and assembly-oriented view of work planners, the sales-oriented view of business person, and other respective views of stakeholders involved in the PDP[9]. Due to the high complexity of PLM caused by tight integration of all components, companies are forced to reconsider traditional work processes and long-established habits. This applies in particular to individual solutions in the lifecycle, which so far has not been required any centralized data basis and collaboration.

3.2.1 Lifecycle Management Approach

In the context of PLM, life cycle management is understood as a concept that divides the process of an object into individual phases of life. For example, an object can be a product, person or project, which can change its state during development phases.

[9] Product development process (PDP) describe the workflows of a product from the product requirements to process planning and sometimes even into production [Cf. ES09, p. 1]

The term "*product life cycle*" originally comes from the Business Administration focusing primarily on the life cycle of products as well as describing the process between the completion and launch up to the removal of a product from the market. These individual stages of a product are sub-divided into sections, into which the product enters or leaves. Based on the perspective, a life cycle can be defined abstractly or on the granular basis. Likewise, life cycles may include subordinate life cycles that are dependent to the main life cycle as well as parallel life cycles and may influence accordingly. This means that all objects have a life cycle and thus complex structures. In this regard, the management of the growing and complex structures of life cycles is an enormous challenge.

In the past and even partially today, PLM is mistakenly understood as a single system. PLM is clearly a concept that provides information from all participating information systems along the value chain during the life cycle in a complete, actual and consistent state. The concept develops its effect only through the combination and cross-communication between the individual solution components. Therefore, PLM can be considered as a building block that can strategically produce individual and situation-specific PLM concepts for the planning and realization by different modules. PLM focuses primarily on the process management and less on the technical systems. Due to the confusion of the people regarding the different used and understood terms, the *Liebensteiner Theses* have been defined.

The following definitions have been adopted as the *Liebensteiner Theses*[10] in 2004 [Sen09, pp. 27–28].

1. PLM is a concept, but it is not a system
   and it is not a (self-contained) solution
2. The implementation and realization of a
   PLM concept requires solution components
3. Solution components are CAD, CAE, CAM, VR,
   PDM and other applications, which support the
   product development process (PDP)
4. Interfaces to other application fields such as ERP,
   SCM, and CRM are also considered as components
   of a PLM concept
5. PLM vendors offer components and/or services
   for the implementation of PLM concepts

PLM concepts are planned over divisions, locations and companies depending on the requirements. Thus, divisions involved must contribute in the product life cycle phases as well as foreign companies have to be considered as an integral partner in the PLM concept. In this regard, there are corresponding solution components depending on the discipline.

[10] Original German passage translated by Jens Michael Hopf:

1. *"PLM ist ein Konzept, kein System und keine (in sich abgeschlossene) Lösung"*
2. *"Zur Umsetzung und Realisierung eines PLM-Konzeptes werden Lösungskomponenten benötigt"*
3. *"Als Lösungskomponenten zählen CAD, CAE, CAM, VR, PDM und andere Applikationen, die beim Produktentstehungsprozess mitwirken"*
4. *"Schnittstellen zu anderen Anwendungsbereichen wie ERP, SCM oder CRM werden ebenfalls als Komponenten eines PLM-Konzeptes betrachtet"*
5. *"PLM-Anbieter offerieren Komponenten und/oder Dienstleistungen zur Umsetzung von PLM Konzepten"*

3.2.2 Product Data Management (PDM)

Product Data Management (PDM) is a concept, but at the same time, it is the central component of a PLM concept. The PDM system provides a wide range of essential features that are necessary to fulfill complex tasks. The product data management has the task to provide all information to all participating groups centrally in a consistent, complete, actual and auditable state which is generated during the product life cycle phases. Data can be, for example, file contents from documents and drawings as well as linked information. The PDM system represents a software solution which can be considered as a type of information backbone. The acronym PDM was formerly used synonymously with EDM (Engineering Data Management) [Cf. ES09, p. 31]. In addition, there were other creations such as cPDM (Collaborative Product Data Management) by CIMdata, ePDM (electronic Product Lifecycle Management) by AMR [Cf. Sen09, p. 28] and VPDM (Virtual Product Definition Management) by Gartner Group [Cf. ES09, p. 37].

3.2.3 PDM Systems

PDM systems have their origin in the 1980s. During that time period, the focus was on the management of technical drawings and related documents which were created by CAD software. In the early 1990s, the linking of documents with master data, structural data, and simple change processes was made based on the requirements of ISO 9001[11] and product liability conditions [Cf. ES09, p. 36]. Today's PDM systems cover a wide range of basic functionalities, see Figure 3.8. Besides the pure document and workflow management, PDM systems include some functionalities that partially require administrative activities and operate transparent as a background task for the user [Cf. ES09, p. 36]. Some functionalities are more or less visible in the corresponding PDM components. The disciplines listed in Figure 3.9 are widely covered by PDM components for supporting the development of the product.

[11] ISO 9001 is a quality management standard that specifies the minimum requirements for a quality management system, which a company must meet in order to offer products and services based on specified regulatory requirements.

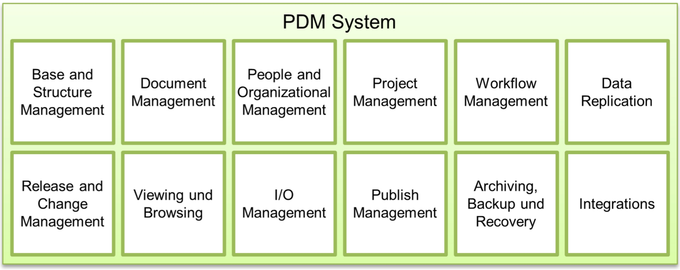


Figure 3.8: Basic functionalities of PDM systems (adapted from [Cf. ES09, p. 36])

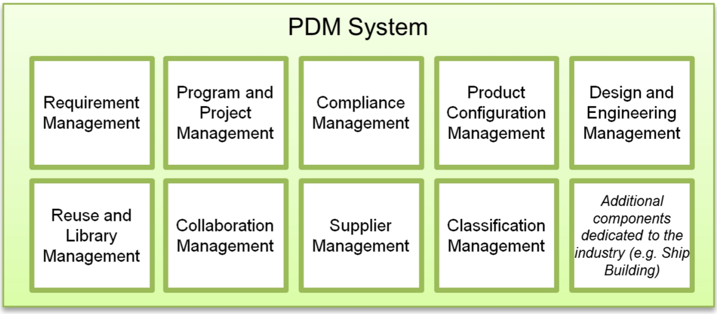


Figure 3.9: Components of PDM systems

Architecture

PDM systems are typically constructed as a three-tier architecture model. This model is subdivided into the data layer, application layer, and client layer, while the interfaces between the layers provide the communication structure. Thus, components from the client layer cannot directly access the data layer, but instead must use component interfaces from the application layer to communicate.

- The data layer holds the data model for all modules of the application layer and provides access to the data resource (such as database or file-based access). The data model includes complex structures and relationships of objects so that PLM vendors provide appropriate tools for adaptation. A direct access to data objects of this layer is prevented by most manufacturers, so that CAx integration components must use the vendor-specific APIs of PDM systems to interact with business objects using API-specific functions and features.

- The application layer contains the business logic to define and control corresponding business processes and workflows. Business objects are used to set information in the respective context to interact with other business objects using relationships and to manage their life cycle. The application layer also provides the interfaces for the integration of CAx solution components. PLM vendors realize the component logic of the application layer with application servers and provide the functionality separately for each component.
- The client tier reflects all client based applications, which allow access to the business logic through a user interface for all users. These include Web browsers and locally installed applications that exchange corresponding information with the application layer using communication protocols.

People and Organization

Through the involvement of all people and organizations (P&O) in the product development, the administrative structure of the access and rights management is an enormous demand on PDM systems. The requests must be carefully regulated. This requires a high granularity of permissions, so that users with associated user roles have access to only necessary information, which is required to perform specific tasks. Therefore, PDM systems must be able to distinguish between organizations, business units, departments, and user roles. The P&O management within a PDM system is an elementary component, because it is crucial for the implementation of security concept. Thus, it becomes possible to limit the access to information for unauthorized persons and to influence life cycles, processes, and workflows only by certain groups of people. Therefore, the P&O is also important to ensure the necessary quality requirements for products. The key concept of the P&O

is to identify the user roles that are necessary for the implementation of projects within the company. The persons in charge of a project obtain assigned appropriate user roles. Moreover, it is possible that individuals play different functions and thus have to assign multiple user roles in the organization. Once a person has logged into the PDM system, the context of the person and the work environment are determined and defined. Some PDM manufacturers (e.g. Dassault Systèmes) have implemented the P&O concept according to the guidelines and principles defined in ISO 17799[12] for information security management in an organization. Every PDM vendor has already a number of predefined default user roles for the PDM system. The user roles are subject to a hierarchical structure. For example, the user role "manager" has more permissions than the user role "viewer". If a person has been assigned to both user roles, the permissions are inherited accordingly. However, the P&O concept does not distinguish whether a user, a role, or an organization perform the respective tasks from a mobile context to meet the user's role. Table C.1 of Appendix C.2 lists some generic user roles of PDM systems.

PDM Components

PDM systems are typically designed modular. The base of a PDM solution provides the basic functionalities which can be enhanced through extensions and industry-specific modules. This kind of enhancement does not represent a customization of a PDM solution but represents the combination of solution modules to a PLM landscape, which can be tailored by configuration and customization to customer specific needs.

Requirement Management

Requirement management is a sub-discipline of the Requirement Engineering (RE), which aims to reach a common understanding of

[12] The control objectives and controls in ISO/IEC 17799:2005 "are intended to be implemented to meet the requirements identified by a risk assessment. The standard is intended as a common basis and practical guideline for developing organizational security standards and effective security management practices, and to help build confidence in interorganizational activities." [Vac09, p. 624]

a conceptual product. In doing so, it is necessary to differentiate between functional requirements (product features, such as a car must have 4 wheels) and non-functional requirements (product characteristics, such as scalability, performance, and maintainability) [Cf. RR06, pp. 9–10]. In addition, the complexity may increase by other requirement dimensions, which are related to the product creation (project requirements) or pursue a corporate strategy (e.g. portfolio management of the company). The V-Model and Rational Unified Process (RUP) have been proven as methodological approaches of the requirements management [Cf. HV04, p. 2]. The Requirement Engineering covers various sub-disciplines which are listed in Table 3.6. Through a variety of publications related to requirements management, a series of definitions based on different perspectives of the fields software development, system development, product development and process development have been defined. Depending on the person's perspective, other aspects will be placed in the foreground. Therefore, a general definition of the term does not exist. Booch[13] defines the term from a systemic perspective, while Balzert[14],

Requirement Engineering
Requirements Definition (task-related actions)
- Requirements analysis (identification) and requirement specification - Requirement validation - Requirement documentation
Requirements Management (action processing)
- Regulation and monitoring (Risk Management) - Controlling (Change Management) - Administration (Implementation Management)

Table 3.6: Sub-disciplines of requirement engineering (adapted from [Cf. MS05] and [Cf. Pre00])

[13] Booch defines the term as follows: "A requirement is a prerequisite or capability that must be met by a system." [Kru99, p. 8]

[14] According to Balzert, the requirement is defined from the product perspective as follows: "Statement about a settled qualitative and/or quantitative characteristic of a product; a fixed system specification of a client to define a system for the developer." [Bal09, p. 589]

DIN[15] and IEEE[16] take the term from a product specific perspective. In context of PLM, this research work defines the requirement for a product according to the following aspects:

- Requirements define specified properties, which give the product capabilities
- Requirements must be verifiable and traceable

Requirements management is not an isolated component of PDM, but is considered as a networked and product-defining component with a variety of relationships to other components. Template management systems[17] help to capture recurring requirements efficiently by predefined templates. Thereby requirements are captured, managed and processed in the PDM system, which allows downstream processes to use these requirements for further processing and to combine it with the product structure. The product structure can be used in turn to trace back defined requirements. However, mobile users are already faced with barriers in capturing requirements on site, because PDM systems usually do not provide direct and intuitive input possibilities for mobile contexts and thus only allow the capturing of text-based content with attachments. The visual and auditory data capturing by an integral PDM component does not exist, so that mobile users are forced to formulate all requirements through text-based descriptions. In addition, the practice has shown that requirements are documented either as a written note described in a spreadsheet or as an idea for later documentation. These detours carry the risk that requirements to be transferred to the PDM

[15] The DIN EN ISO 9000:2005 has defined other definition: The requirement refers to "need or expectation that is stated, generally implied or obligatory" [Deu12]

[16] According to the standard of the Institute of Electrical and Electronic Engineers (IEEE) 610.12-1990 is a documented requirement defined as follows [IEE90, p. 172]

1. *A condition or capability needed by a user to solve a problem or achieve an objective.*
2. *A condition or capability that must be met or possessed by a product or product component to satisfy a contract, standard, specification, or other formally imposed documents.*
3. *A documented representation of a condition or capability as in (1) or (2).*

[17] The task of template management systems is the collection of elements (such as attributes and methods) to a template which is dynamically re-usable for use cases.

system at a later point in time are incomplete and incorrect and therefore cannot be interpreted correctly anymore. A consistent set of requirements management should prevent these kinds of inconsistencies and errors during the product development. The requirements are categorized and managed for specific topics and controlled by approval workflows (see Figure 3.10)). Different actors in the PDM system (e.g. creator, tester and leader) fulfill their associated roles in requirements engineering to control formal release and change processes. The P&O of a PDM system supports the implementation of the role and rights concept. In the future, the requirements management will increasingly be coupled with the innovation management[18] and idea management[19] based on the implementation of the corporate strategy, because the results of these two areas are closely related to requirements which are to be defined and managed in accordance [Cf. Val11, p. 16]. Especially idea management requires the creativity of employees. In view of the proceeding and profound change to intelligent and networked products in context of IoT[20], the competitive pressure for companies rises to create innovative smart products to obtain the ability to innovate. The key to innovation lies particularly in the perception of engineers in how real objects interact with the surrounded environment. Therefore, free space for the development of innovation must be given to the engineers to minimize the information gap in capturing ideas and requirements between the real and virtual world. The barrier of information capturing which is a cause of the lack of information can be overcome by the use of mobile features.

Program and Project Management

The center of each life cycle in PLM is represented by business objects which are subject to continuous change processes by persons or

[18] Innovation management is understood as the system-based planning, management and control of innovation in companies. Innovation process models such as stage-gate model applies different stages for brainstorming and idea evaluation, product development, product distribution as well as product marketing [Cf. VB05].

[19] The idea management covers the generation, collection, and evaluating of ideas for new or improved product property. The idea management supports the stimulation of the idea-finding process through a creative and social working environment.

[20] Internet of Things (IoT)

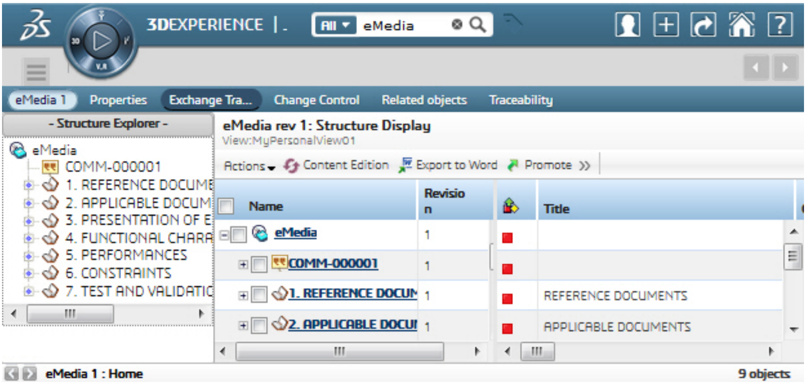


Figure 3.10: Web-based client for requirements capturing in ENOVIA

organizations. Business objects can represent drawings, single documents as well as components and assemblies. To ensure that all business objects with respective changes, group of persons involved, constraints and available resources are manageable, business objects are organized and structured into a project which will be consolidated within a program due to the large number of projects and subprojects. The primary duties of the program and project management in PDM systems include the functionalities listed in Table 3.7. The project management within PDM systems has the advantage that the full and complete integration of PDM components allows the access to all data compare to traditional project management tools such as Microsoft Project. The project management uses the core functionality of the PDM system, for example by adding and removing users and groups to/from a project instead of providing a separate user management. Likewise, there is a close link to other PDM modules by using PLM information from the requirements management and materials management for the project management. The project represent a frame to structure all PLM information process-oriented in the execution of a process and assigns granular tasks to appropriate persons [Cf. ES09, p. 170]. In addition, most of the PDM vendors offer integration for third-party software to carry out the project management.

Project Management Functionalities
Core functionalities
- Defining the business objectives - Definition of milestones - Determination of quality criteria - Content and workflow management within projects - Structured and complex task management and tracking - Managing project information and status publishing
Advanced functionalities
- Managing meetings and documentation of key decisions - Project templates for recurring projects - Overview about programs and projects from a management perspective - Risk management - Financial planning and monitoring - Resource planning

Table 3.7: Project management functionalities of PDM systems

Such external project management tools rarely reach the required level of integration, so that not all necessary information is available and thus additional effort occurs through customization. In addition to the task management, material costs, individual operation costs, quality criteria, project risks, and resource planning can be considered in the project. The project overview provides the management with the possibility to check the status of the respective phase of projects and thus respond quickly to issues. The visualized project views represent real-time information from the PDM system.

Despite the extensive project management capabilities, the various media content have not been sufficiently taken into account. Hence, an integration of mobile features does not exist, so that users have to utilize third-party tools for the creation of voice-based notes, pictures, videos and other media content. Subsequently, the media content can only be added as an attachment to the project (see Figure 3.11). Therefore, these detours for the medial contribution are hardly accepted by stationary and mobile users, because the enormous required time for creation, transformation, and transmission of captured content is not in relation to user's personal benefit. In addition to the existing text-based content, media content has to assist the user in the interpretation of information through various forms of perception (e.g. visual and auditory perception), the identification of problems and the generation of creative ideas. However, the creation of content represents a barrier that must be overcome. In order to remove this limitation in the project management of PDM systems, a direct integration of media tools must exist. This objective can be achieved by the integration of mobile features.

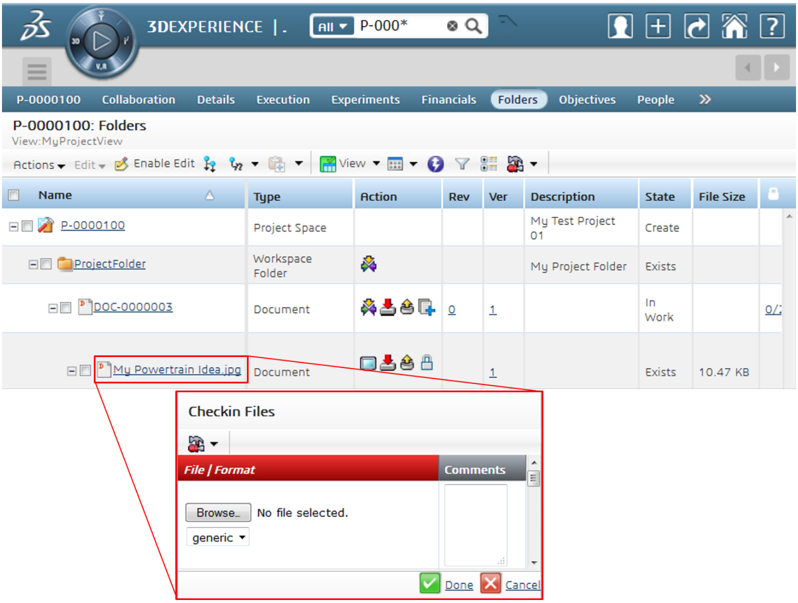


Figure 3.11: Adding media content to a project in ENOVIA

Product Configuration Management

According to the definition of ISO 10007:2003, the configuration management is understood as "[c]oordinated activities to direct and control configuration". The configuration refers to the "[i]nterrelated functional and physical characteristics of a product defined in product configuration information". The information of a product configuration represents "[r]equirements for product design, realization, verification, operation and support". The configuration item refers to an "[e]ntity within a configuration that satisfies an end use function" [Int09]. A *Configuration Item* (CI) can be any resource whose existence and version was captured [Cf. BvB04, p. 76]. The configuration specifies the product and its characteristics at a specific time. This specification represents a snapshot that contains a collection of information necessary for the manufacture, installation, maintenance, and quality management [Cf. ES09, p. 113]. Therefore, the configuration must not reflect the latest product status, but rather a valid and coordinated configuration of all objects involved. The configuration can include, for example, product structures, engineering designs, software (embedded software), simulation results, numerical control programs, and manuals.

From the management perspective, the Configuration Management controls and regulates the development of a product by the identification of product components and the suggestive control of changes. According to ISO 10007:2003, the Configuration Management is divided into the four sub-activities: Configuration Identification, Configuration Control, Configuration Status Accounting, and Configuration Audit [Cf. Int09]. Knowledge capture takes place in a text-based form compared to other PDM components. File-based content is added as an attachment to an existing object (e.g. document object). Once the content has been added to the PDM system, social and collaborative interactions can be performed between users. An integral component for media content does not exist for PDM systems, so that users have to create the media content with tools from third party vendors. Subsequently, the generated file-based content of the third-party tool must be manually attached to the corresponding object of the PDM system by the user.

In addition, the Variant Management as a subdiscipline of the Product Configuration Management does not provide an opportunity to discuss variants of a product or product family without creating the necessary objects and configurations in the PDM system. For this purpose, it would be necessary to have a kind of mental scratchpad for ideas that have not been formulated. Thereby, creative ideas about product variants could be shared and actively discussed early in a social and collaborative environment. In practice, the discussion functions of a PDM system are rarely used by (mobile) users because of a missing creative space of communication as well as the perception of users who have the impression that information stored in the PDM system are finalized data and thus information (private discussions and images) cannot be deleted anymore. These reasons have to be especially investigated from a social point of view, because this perception of users represents a blockade for the individual creativity. Mobile features can help to resolve this issue by managing mental notes of mobile users medially and to enable social interactions outside of PDM systems. Thereby, the PDM systems would support social interaction between users with PLM information. Subsequently, the result of social interactions (e.g. concrete and partially formulated ideas of a product variation) could be synchronized with the PDM system by creating the corresponding objects in the PDM system. Mobile features would act as a link between the Idea Management and the Variant Management as well as support the early phase of the Configuration Management.

Engineering Configuration Management

The term Engineering Change Management (ECM) is generally understood as the discipline in order to organize product changes through processes and people involved. The SASIG defines the term as "the coordinated management and uniform tracking of Engineering Changes, starting with the identification of potential for change and ending with the manufacturing implementation of change" [Str10, p. 6]. A change is understood as a modification on a copy of an existing part, while the original part itself remains unchanged. A change can represent any kind of adaptation, which can have impacts on the shape and function of a part or the whole product. Moreover, changes can also exist within a complex structure of changes between which have interactions and dependencies. The necessity for the implementation of changes in context of products is essential and represents a normal process. It is inevitable that new product demands require adaptations of the product through modifications. Likewise, a product change can not only cost reduction and error correction, but also can be triggered by new technologies, standards, and changes in the legal framework conditions. Engineering Change Management (ECM) occupies a significant position in industry. In this context, Engineering Changes (ECs) are mainly used for improvements and bug fixes of parts as well as the entire product. In 2005, the number of internal ECs was estimated by DaimlerChrysler, Ford, and General Motors at approximately 350 000 ECs per year with cost over $ 50 000 per EC [Cf. WSV11, p. 533]. This indicates that ECs have a huge impact on product costs and thus affect the competitiveness of the company. The cost of ECs largely depends on the scope and development stage of the product in which the EC is created and processed. The cost of ECs is significantly lower in early stages of the development, because fewer people need to be involved in the change process [Cf. Sta03, p. 297]. In case that an EC is necessary after the product has already been promoted to the operation and service phase, a product recall must be eventually performed by the company. This can result in dramatic cost increases and a negative image effects for the manufacturer. The Change Management can develop complex structures, which lead to difficulties for people involved. An EC can be negatively influenced by clerical errors, disastrous coordination between all the people involved, wrong interpretation of decisions, and simply documentation lack for any decision taken. Negative impacts on ECs are explained by the automotive industry through the following primary causes [Cf. Kin09, p. 1]:

- Multiple systems and formats
- Missing information
- Wait time and responsiveness
- Manual re-keying of information
- Translation and interpretation
- Conflicting changes

The consequences that result from these negative problems could be devastating. It may lead to an increase of the planned duration for EC implementations, ECs cannot be implemented anymore or ECs have negative impacts on functions and performance of the product or related components. Different engineering change processes have been proposed by Dale [Dal82], Maull et al. [MHB92] and Jarratt et al. [CE05]. The proposed approaches differ in scope, number, and division of respective phases. The problems mentioned above can partially be solved by a close collaboration between persons involved. Because some group of people cannot constantly monitor the latest problems using a stationary workstation, they are temporary excluded from the information update when they are moving (see Figure 3.12). This issue can cause long waiting times and responsiveness leading to delays of underlying processes. Only after recovering the connectivity with the PDM system, the information gap can be closed by the person. This applies on the one hand to the information carrier which has to communicate the information to the PDM system, but on the other hand to the people involved in retrieving the information from the PDM system. From an abstract perspective, this represents the classic publish-subscribe design pattern by the GoF[21]. It also applies here to avoid the manual re-entry and incorrect interpretation of information by using mobile features as an integral component of the PDM system.

Reuse and Library Management

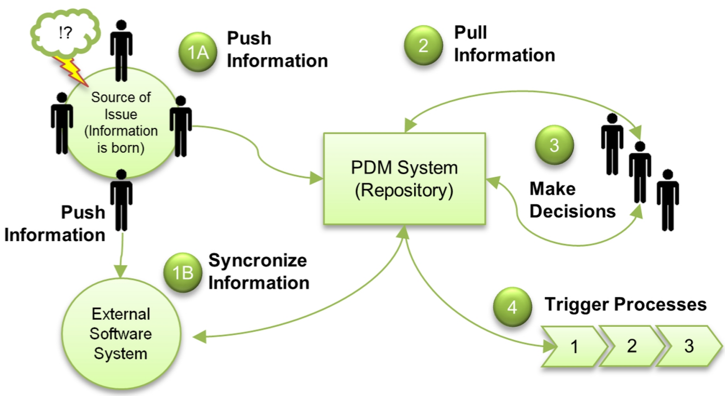


Figure 3.12: Generic information flow

[21] Gang of Four (GoF) refers to the four persons Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, who define a set of reusable design patterns that have been established in software engineering.

Libraries represent an important module for companies for the reusable generated product content. The knowledge that has been generated in the phases of product development is stored by a variety of information such as documents, technical drawings, parts, and assemblies. The objective of libraries is to provide information already generated in a structured and categorized form to use it again for new products. Thereby, it is intended that the productivity of companies be boosted by avoiding the development costs for new components and significantly reducing the duration of product development. It is therefore all the more important to store information intelligently in the PDM system for ensuring the fast and intuitive retrieval of information. Classification systems have been established as an important method for the structuring of knowledge in context of PDM. The objective of classification systems is to convert unstructured knowledge into a structured knowledge network through a principle of order based on characteristics. Organisational systems can be divided into three areas listed in Table 3.8. The geometrical classification search system searches for similar objects based on a reference part which has similar geometrical structures. To ensure that new objects can be found, they are initially converted into a neutral data format, such as JT, STL or VRML [Cf. ES09, p. 160]. The geometrical search system determines the thumbnails from the data to simplify the visual representation as well as the respective CAD geometry of parts. The search results are ranked and sorted according to geometrically similar

Organisational Systems
Hierarchical Systems
- Class list of characteristics - Geometric classification
Non-Hierarchical Systems
- Full text search - Indexed full text search (Index) - Tags
Partial Hierarchical Systems
- Thesaurus

Table 3.8: Organisational systems (adapted from [Cf. Kra03, pp. 37–38], [Cf. Eig91], [Cf. Enk78], and [Cf. Mew73])

geometries, which are represented visually by thumbnails. Geometric grouping is not considered in this classification type. In this regard, there are already a couple of vendors such as GEOsearch by CADENAS and SIMILIA by SIMUFORM. However, no integral functionality for (mobile) user exists to perform a geometric similarity search based on a physical object from the real world. Although the user can create an image of the real reference part using a third party utility, the user must subsequently copy the captured image from the camera to the workstation, transform the image, and upload them to the search engine. Only after completing all previous steps, the geometric similarity search can be performed. These steps are too cumbersome for the majority of users. Therefore, a possibility must be created to use the physical object directly as a reference for the geometric similarity search. Mobile features as an integral component of the PDM system could solve this issue by reducing the mandatory steps of the geometric similarity search to a minimum. Therefore, it is necessary to analyze the captured image to detect the contours of the real object and to provide the results to the search engine for processing.

Engineering Collaboration Management

For complex PLM system solutions, it is essential to support the collaboration of distributed teams. In this regard, two different approaches have been established to promote the needs of a distributed collaboration. The concepts are based on the requirements of collaboration. The first concept concentrates on the promotion of data exchange between external partners, while the second concept focuses on the collaboration of groups using the same data repository. The second concept refers to the research field of Computer Supported Cooperative Work (CSCW)[22], which is seen as an interdisciplinary field of research in computer science, sociology, and other disciplines. CSCW deals with the fundamental collaboration of people under consideration of social aspects through the use of technologies. Due to the different areas of disciplines, totally

[22] Bowers and Benford defines CSCW as follows: "[...] [I]n its most general form, CSCW examines the possibilities and effects of technological support for humans involved in collaborative group communication and work processes" [BB91, p. V]

different and specific perspectives are taken into consideration for CSCW concepts, which are included in the development of new technologies in order to achieve an appropriate level of user acceptance. In this context, a particular challenge represents the access management, which can spread through projects and organizations. Due to the focus on mobile feature, only the CSCW is studied in this research work. Favourable conditions must be firstly created to realize collaboration. Therefore, it is necessary to implement an appropriate concept for building virtual environments (also referred as collaboration room [Cf. Gri06, p. 172], collaborative space, or workspace).

The virtual work environment is usually set up to allow all people to access the same database, regardless of their location. However, this conditionally applies only for mobile users, so that virtual work environments are not available in all functionalities as it is for stationary users. The provision of the data repository takes place directly in the PLM environment or in a separate collaboration server. Integrated collaboration tools provided by PDM solutions have the advantage that the access to data objects for each person is directly controlled by the P&O. Separate collaboration solutions require higher administrative efforts through the administration of users, access control, and workflows. The P&O of the PDM system ensures that data objects can only be accessed with the necessary collaboration and fulfillment of tasks. The consideration of mobile users with characteristics such as location of access and type of access device has not yet taken place, because there are currently not well-developed concept of mobile users.

The collaboration in a stationary environment is characterized by synchronous and asynchronous forms of communication [Cf. GK09, p. 79]. Asynchronous communication takes place in time independent cooperation. This means that the performed interaction of an employee is perceived at different times by other employees. Employees are informed about necessary tasks to be performed by notifications (e.g. by e-mail or SMS), which were triggered by interactions of other employees or systems (e.g. IconMail in the PDM system ENOVIA V6). The synchronous communication allows interactions between multiple employees at the same time. The focal point is a data object (e.g. document, drawing, part) that is processed together with the support of collaboration tools. The most commonly used collaboration tools are instant messaging and video conferencing from the cluster of groupware software. PDM vendors increasingly tend to provide appropriate functionalities of collaboration with integrated software components in PDM systems. Continuous data flows and more efficient workflows between the data backbone and collaboration components can be achieved by such integral components. Although the group of mobile users is steadily growing in companies, these mobile collaboration tools, however, are barely taken into account in the PLM environment. This leads to the fact that mobile users are excluded from the collective cooperation. Moreover, it is desirable to use mobile collaboration components for the information exchange, which does not place the existing PLM object (e.g. drawing, document) in the foreground to provide a creative space for mobile users. This can be for example an exchange of sketches, video sequences, and images which have first once nothing to do with the PLM objects. However, a subsequent transfer of relevant information to the PDM system is imperative, once considerations are captured and developed at a later stage into more fully specified ideas to comply with the concept of a common data repository.

Compliance Management

Compliance Management deals with the definition and compliance of requirements for a product or service (regulations) under consideration of various aspects. One aspect of the product depends on the regulatory requirements, which can be for example exports (e.g. ITAR[23]), ecological regulations (e.g. ELV[24], RoHS[25], REACH[26]), document management to

[23] International Traffic in Arms Regulations (ITAR) is an American regulatory framework to control export of weapons, armaments and defense equipment

[24] End of Life Vehicles (ELV) is an EU directive for the material recovery of end of life vehicles by recycling

[25] Restriction of (the use of Certain) Hazardous Substances (RoHS) is a European Union directive which regulates the handling of hazardous substances in electrical and electronic equipment

[26] Registration, Evaluation, Authorization and Restriction of Chemicals (REACH) is a European Union regulation for the transport of chemicals

comply with archiving regulations (e.g. U.S. 5015.2-STD[27]) or medical regulations (e.g. FDA[28]). The intention is to ensure that products and processes work in accordance with rules and regulations as well as ensure traceability. The number and complexity of country-specific legal regulations continues to grow, so that the challenge to develop compliant products increases by higher environmental protection requirements, a growing need for customers safety and, other reasons. A violation of statutory provisions may have the consequence that the cost of product development rises by delays or fines, which cause competitive disadvantage or in the worst-case scenario, an expulsion from major regulated markets. Efficient management and control structures are necessary and must be provided by the PDM system to comply with legal provisions for a product.

The subject can be divided into the areas: "*binding compliance*" and "*conformity*" [Cf. Sta07, p. 173]. The first area deals with the compliance of legally prescribed and mandatory requirements that are imposed and must be complied by the product. The second area addresses voluntarily specified requirements to products or services that are subject to respect by the company to live certain social, moral, and ethical aspects or values, which are based on the corporate culture. These self-imposed rules are summarized under the term *Code of Conduct*. The validation of defined requirements for a product takes place separately. This means that the respective process exists in the PDM system, but the practical implementation takes place independently in appropriate laboratories. In this case, various problems may arise between defined requirements in the PDM system and the practical implementation. For example, information from the PDM system can be misinterpreted by the user and errors occur through re-entering information manually to the device. Both problems are avoidable through an integral solution by connecting the device for the validation with the requirements in the PDM system as well as establishing a communication link between both systems. This requires

27 United States Department of Defense (U.S. DoD) 5015.2-STD is a norm for compliance with requirements of the management of records within the Department of Defense which is used by many U.S. federal, state and local governments

28 Food and Drug Administration (FDA) is an American agency for food monitoring and pharmaceutical approvals

intelligent and networked devices (commonly known as smart products) that operate in context of IoT. Mobile features can close the information gap with an integral component of the PDM system as well as act as a bridge between the PLM world and smart devices. Furthermore, there is a wide variety of other industry-specific PDM components addressing the needs of various industries. A selection of industries with specific requirements to PLM concepts is described in Appendix C.1.

3.3 PLM and MICT

The last two sections have covered the themes Mobile ICT and PLM separately. The admission of mobile users to access stationary PDM systems using mobile features requires the consideration of both topics together. Therefore, mobile PLM solutions are examined and analyzed to get a deeper understanding of technical, social, and business issues regarding mobile ICT solutions in context of PLM. Moreover, key aspects of mobile PLM solutions are addressed and their benefits are evaluated.

3.3.1 Usage Scenarios in the Mobile Field

Considering various PLM concepts, it is necessary to distinguish between stationary and mobile PLM Services as well as between stationary and mobile PLM Clients. PLM Services describe provided services that can be used and consumed by PLM Clients. Based on the matrix of possible PLM scenarios (see Table 3.9), a whole serie of different solutions arises. Classic PLM Services (SS) cover common applications, in which a stationary PLM Client (such as a workstation) has a reliable access to stationary PLM Services using the company's internal network. This category also includes conventional CAD workstations of Design Engineers. The category of mobile PLM Services (SM) plays only a minor role in the market, because the performance and benefits are limited on mobile systems. The SM-category includes use cases that describe the access of stationary PLM Clients on mobile PLM Services or components of partially autonomous PLM Services. The group Mobile Service (MS) describe scenarios that access stationary PLM Services using mobile

PLM Clients (mobile devices such as smartphones and tablet). Most available approaches and solutions on the market are represented by this category. This work examines the use cases of Mobile Services category to identify relevant patterns, abstract and group them as well as implement into the Mobile Feature Framework. The last group of Peer-to-Peer Services describes scenarios of direct interaction and communication between PLM Services and PLM Clients. In addition, the group also includes applications of direct communications between PLM Clients without the involvement of a corresponding PLM Service component.

		PLM Clients	
PLM Service	**Constellation**	**Stationary**	**Mobile**
	Stationary	Classic PLM-Services (SS)	Mobile Services (MS)
	Mobile	Mobile PLM-Services (SM)	Peer-to-Peer Services (MM)

Table 3.9: PLM scenarios for service and clients (adapted from [Cf. Bul08, p. 55] (qtd. in Hampe and Schwabe, p. 304))

The benefits of Mobile Services for PLM are particularly suitable for company employees who have to perform time-dependent and site-specific tasks. Time-dependent information is a set of data that must be available at a certain time to continue activities (e.g. task, discussion). Location-specific information is a piece of data that has a relationship to the geographical location of the person commonly named location-based information. Hess discovered in his study of CRM applications that a need for CRM applications exists especially for employees who require a wide range of time-critical and location-specific information [Cf. HFH^{+}05, p. 12]. The demand for mobile PLM applications that provides time-critical and location-specific information has been also seen in this work. However, reality presents a different picture. In a survey by Christmann and Rohmann in 2012, only 33% of the companies have strongly or very strongly involved mobile devices in stationary processes and half of the companies have only used standard services such as accessing the Internet access or e-mail [Cf. Chr12, pp. 116–127]. Table 3.10 contains examples of the information flow between PLM and employees in a mobile context.

	From PLM to employees	**From employee to PLM**
Time-dependent Information	Obtain information (e.g. product information) and decision making	Communicate decisions, discuss information (e.g. requirements)
Location-dependent Information	Information needs, location determination and location-based radius search of objects, navigation to objects, calculation of location-dependent dates, information reduction (filtering), and information visualization	Capturing of multimedia content, and transmission of captured information

Table 3.10: Time and location dependent information flow between PLM and employees (adapted from [Cf. Bul08, p. 56])

3.3.2 Mobile PLM Applications

Due to the enormously successful market penetration of mobile devices (such as smartphones and tablet PCs) based on the introduction of the iPhone, the socket for mobile PLM applications was created. This section examines the particularities of mobile applications and analyzes the following aspects: The architecture of mobile applications[29], differently used forms of communication, and the integration of employees in the PLM concept especially in workflows and business processes.

[29] The architecture of mobile operating systems is discussed in Section 3.1.3

Architecture of mobile applications

Mobile applications can run on a range of mobile devices. It is important to distinguish whether mobile devices are used exclusively for visual representations of the user interface or for running applications on mobile devices in order to access hardware functions as well as web content locally or remotely from a server instance.

- **Group 1: Remote display applications** transmit the screen contents of a desktop operating system or individual applications over the network using appropriate (usually proprietary) network protocols such as RDP[30] or ICA[31]. In this market segment, vendors such as Citrix and NoMachine[32] are specialized to provide appropriate configurable applications for users. The architecture requires local clients who perform the access operation as well as compression and encryption of the data flow. The access to hardware feature of the mobile operating system is limited for server-side running applications. In addition, the use of the server provided application without network connectivity is not possible.
- **Group 2: Web-based applications** run on the server such as remote display applications, but the access to web content is done by native web browser applications, which establish appropriate connections to the PLM Service. The PLM services run on a server instance. Only web content (e.g. forms, tables, and images) is rendered in the web browser and visually presented to the user. Such browser-based applications are called web applications that use communication

[30] Remote Desktop Protocol (RDP) is a proprietary network protocol from Microsoft that is used for remote maintenance of computers. RDP is based on the reliable transmission control protocol (TCP). By default, the service is provided on port 3389.

[31] Independent Computing Architecture (ICA) is a proprietary network protocol from Citrix Systems, which provides comparable basic functions such as the remote desktop protocol, but in particular offers additional access to hardware interfaces (e.g. local scanning, remote USB) and features (e.g. rotation of 3D objects, use with narrowband network connections). By default, the ICA service is provided on port 1494.

[32] NoMachine NX is a proprietary software for transferring screen contents. The software compresses the X11 data flow between the client and the server to reduce the total amount of transmitted data.

protocols such as HTTP[33] or HTTPS[34]. Web apps always require a network link and have very limited access to hardware features of the mobile operating system. The major advantage of this technology is the source code portability for other mobile platforms.

- **Group 3: Native applications** are full fledged applications and completely executed on mobile devices and provide to the users a wide range of native features and functions compare to pure UI visualization technologies or web-applications. Such extensions (e.g. store changes locally, offline mode and access to hardware features) may also collect the user context and react to the context accordingly. Native applications are bound to a multi-layer model, which was explained in Section 3.1.3. However, the use of native functions represents a challenge to the portability of the source code to other mobile platforms. Therefore it is necessary to develop native applications for each respective mobile platform separately, which in turn increases the development effort significantly.
- **Group 4: Hybrid applications** combine the advantages of web-based and native applications by a native container which provides access to platform specific native capabilities, whereas the web content is presented through an embedded native web browser. However, a container must be implemented for each mobile platform. Native web browser with various support forms of web standards represents an additional challenge in the centralized provision of web content for various mobile platforms. However, hybrid applications can provide enhanced functionality, which takes the user context into account and saves web content locally when no network connection is available. Appendix B.7 summarizes in Table B.4 the characteristics of respective architectures and the resulting advantages and disadvantages.

[33] Hypertext Transfer Protocol (HTTP) is a TCP-based transmission protocol for displaying web content which by default provides the service on port 80.

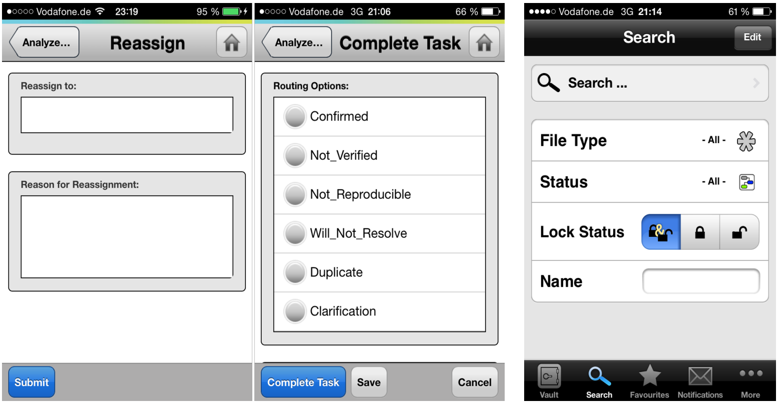
[34] Hypertext Transfer Protocol Secure (HTTPS) is a communications protocol for secure communication, which is identical to the scheme of HTTP, but used for encryption as an extension of the transport layer settled Transport Layer Security (TLS) protocol (originally known as Secure Sockets Layer (SSL))

Examples of mobile PLM applications

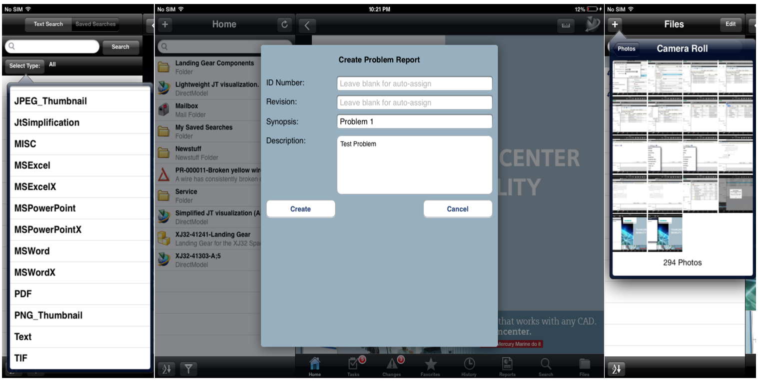
The most manufacturers of PLM applications for mobile platforms focus on web-based applications and hybrid applications. Especially in the early days of mobile PLM world, native web browser apps have been used to load the common web-based user interfaces. However, manufacturers had failed to customize the user interface for smaller mobile screen sizes, so that the user experience was negatively affected by unnecessary conditional scrolling and small font sizes. This led that manufacturers are increasingly required to adapt the user interfaces for mobile devices, which were originally designed for stationary workstations.

The manufacturer Aras is particularly focused with the product *Porchys* on the advantages of hybrid application to use various mobile platforms like Android, iOS, BlackBerry [Cf. Poo11]. Porchys allows the access to project information such as status, milestones, and scheduling as well as the management of customer and compliance requirements. PTC offers with the product *Windchill Mobile* [Cf. PTC13] two possibilities to perform the access to the PDM system *PDMLink data*. A native application for Android and iOS allows basic tasks such as search queries (see Figure 3.13a), retrieving documents, and product information as well as workflow activities. The manufacturer refers to conventional user interface using a Webkit-enabled mobile web browser to apply the full PDM functionality with certain restrictions which may have led to the mentioned issues from the early days of mobile PLM world[35]. The mobile application *mobilePDM* manufactured by Dassault Systèmes provides a native application for the mobile platform iOS to retrieve information from the *SolidWorks Enterprise PDM* System. This application allows the reading of document content, searching for various file types (see Figure 3.13b), and multimedia files that have been stored in the PDM system. In addition, content from the PDM system can be stored locally, in case the network connection is unavailable [Cf. Das12]. The manufacturer Siemens offers with the product *Teamcenter Mobility* a native application for the mobile platform iOS that accesses information from the PDM system *Teamcenter* and visualizes 3D models locally (see Figure 3.13c).

[35] For example, the execution of Java applets not possible on mobile devices



a) *Task actions in Windchill Mobile v1.2* **b)** *Search in mobilePDM 2.0*



c) *Search for types, create reports, and attach images in Teamcenter Mobility v3.2.1*

Figure 3.13: Mobile PLM apps

Moreover, the manufacturer provides additional features such as offline cache for files and a local JT Viewer as well as the possibility (e.g. release management) to perform activities in the offline mode and to synchronize the changes to the PDM system at a later date, as soon as the network connectivity was recovered [Cf. Sie13]. In general, the majority of PLM applications have a number of similar basic components that can be derived to a corresponding architecture. The basic components of an application include the management of settings, a module for network communications, predefined user masks to perform text-based queries, sorting functions, workflow and change management, management of tasks and changes, activity tracking and logging, modules for the visualization of 3D models (e.g. JT viewer), management components for offline content as well as client modules for the login and session management. Figure 3.14 illustrates schematically the interaction between basic components of native PLM applications.

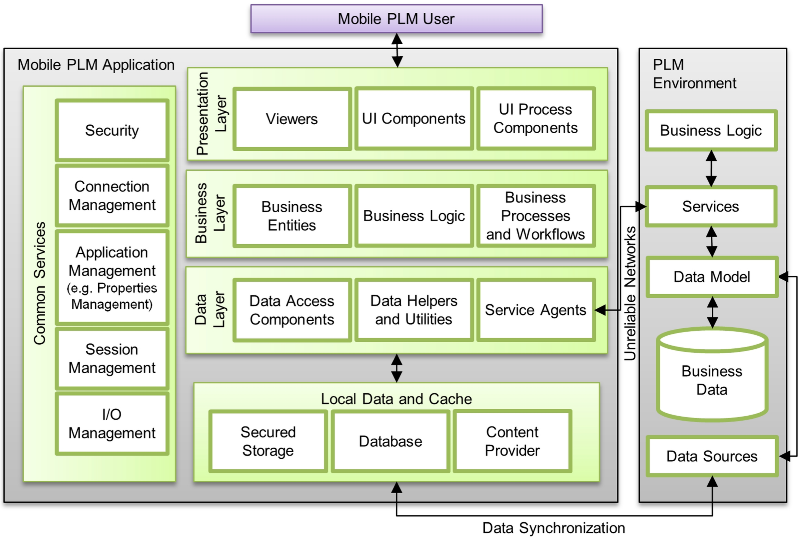


Figure 3.14: Mobile application architecture (adapted from [Cf. Mic09])

3.3.3 Communication Forms of Mobile Applications

Most types of PLM applications are using for the communication the transmission protocols HTTP and HTTPS. The query of PLM information is the primary focus of the user. The consideration of mobile contexts (e.g. location-specific information), mobile feature-based interactions (e.g. voice-based PLM searches, part identification, and gestures control), and media contribution (e.g. voice messages, images, and video capturing) does not take place. The input of information takes place mainly via predefined and is adapted for text-intensive forms that are structurally very similar to input screens of conventional user interfaces (see Figure 3.13a, 3.13b, and 3.13c), but have been adapted for hybrid applications to appropriate screen sizes. The offline functionality of some applications allows users to perform approvals without network connectivity[36], but this also represents a challenge when events with approval already grant have to be synchronized at a later date with the PDM system. Therefore, the application and the PDM system must cooperate closely in order to avoid inconsistent states when several people work on the same object (tasks, parts, etc.) but using different ways. In addition, collaboration tools are missing in the area of mobile PLM, which allows mobile users verbal communication possibilities, collective interaction and manipulation options, and collaboration management. Users have to use separate solutions from other manufacturers. The integration of existing solutions would be beneficial for users to support mobile CSCW[37].

3.3.4 Integration of Employees into the PLM Concept

PDM systems usually provide possibilities to use the existing user account and assigned roles in order to allow users to access PLM information in mobile contexts. Depending on the manufacturer, additional licenses for mobile use are necessary. A server-side management of mobile devices and proactive actions on mobile devices (e.g. locking in case of theft) do not take place. In addition, not all functions are reflected in native applications, as is the case with common web-based user interfaces. In fact, only the provision of basic functions supports the

[36] This option is available in Teamcenter for design verification [Cf. Sie11, p. 6].

[37] Mobile Computer Supported Cooperative Work (MCSCW) describes the collaboration of people who are supported by collaboration tools using mobile information technology

consumption of simplified PLM information (e.g. reports and product information). Thus, mobile users are not always able to fully exercise task with capabilities of assigned roles in mobile situations. Due to technical limitations (e.g. small screen sizes), the mapping of all functionalities for the mobile context is not always required.

In this context, the corresponding use case should be examined more closely and a corresponding mobile solution has to be developed. It should be noted that user contexts are rarely considered, even although the capabilities for native applications exist. In this case, there is enormous potential for mobile PLM applications in the identification of user contexts using local hardware features as well as in the improvement of the user experience for new and existing applications.

3.4 Frameworks for Mobile Applications

A framework itself does not represent an application, but is instead a collection of recurring design patterns that are conceptual structured, abstracted, and collected in a component-based generic architecture. There- by, a framework represents a professionally completed and independently usable, and consistent model. A framework is available to software developers in application programming to reduce costs and expenses (time and personnel) by reusing framework components as well as improve software quality. In addition, the complexity and thus the scale of application can be reduced. Quality assurance activities in frameworks also reduce the error rate of applications compared to fully in-house software solutions. The distinction between the terms framework, library, and toolkit is not well defined. From a general point of view, a framework includes a rule set, which clearly defines the behavior by patterns and specify the usage of features through generic code structures, while toolkits and libraries are focused on the collection of programs, features, and tools [Cf. Ste10, p. 802]. The use of frameworks generally affects the structural design of applications and the software architecture of applications, because the use of libraries and components of the framework requires the compliance of specified design and guideline rules. The guidelines describe the data flow and interfaces that must be complied by applications. Frameworks need to be flexible and expandable enough to keep future adaptations in applications at a minimum when framework components and interfaces are changing. Otherwise, small changes (e.g. interface definition) could cause enormous costs for the adaptation of applications.

3.4.1 Types of Frameworks

Frameworks address different aspects and can be assigned to multiple framework types, so that a strict distinction exists only conditionally between different framework types such as application and domain frameworks, class and test frameworks, and web frameworks. A strict distinction between framework types exists only conditionally, because frameworks can be assigned to multiple framework types.

- **Application frameworks** reflect frame structures that are used by a specific application domain. Thereby, the provided framework structures are required by all applications. Popular frameworks of this class are MFC[38] by Microsoft, Spring[39], and Oracle's ADF[40].
- **Class frameworks** summarize abstractly defined classes and methods that can be used on a wide range of applications.
- **Domain frameworks** do not address common required class structures needed by applications, but whole problem fields of a specific domain with those applications are confronted to fulfill a specific practical purpose. Hereby, use cases are mapped by recurrent design patterns in the framework to simplify the generic application-oriented implementation.

[38] Microsoft Foundation Classes (MFC) is a collection of object-oriented class libraries that are used for the development of graphical user interfaces for C++ applications.

[39] Spring is a framework based on Java to promote generally applicable and accepted programming practices by applying holistic solutions.

[40] Oracle Application Development Framework (Oracle ADF)

- **Testing frameworks** are used as a quality assurance tool to facilitate the automation of software testing. Common frameworks in this category are JUnit[41], Robot Framework[42], and Selenium[43].
- **Mobile web-based application frameworks** are specially designed for the creation of applications that are developed for mobile devices. The frameworks in this class usually combine various disciplines such as JavaScript, HTML, and CSS to reduce dependencies on device-specific applications and native programming languages such as Objective-C. Native dependencies exist, even if device-specific mobile features need to be addressed. The access to mobile features is fully managed by the framework, so that interactions between mobile features and the framework are carried out transparently. Frameworks included in this group are known as Apache Cordova[44], jQuery Mobile[45], Sencha Touch[46], Appcelerator Titanium[47], Worklight[48], and a dozen other frameworks. Thereby, frameworks differ mainly in the functionality, support for mobile platforms, and method of accessing mobile features. In this work, the focus will be on mobile web-based application frameworks. Other framework types and classes are not further investigated.

41 JUnit is a testing framework for applications developed in the Java programming language that is specifically designed for classes and methods testing (so-called Unit).
42 Robot Framework is a generic test framework with a focus on acceptance testing and acceptance test-driven development (ATDD) for testing web applications
43 Selenium by ThoughtWorks is a testing framework for the automated testing
44 Apache Cordova (formerly known as PhoneGap and Apache Callback [Cf. SM14, p. 46]) is a web-based mobile framework with native implementation for accessing platform-specific mobile features
45 jQuery Mobile is a web application framework, which is developed for touch-optimized applications. jQuery Mobile is often used in conjunction with supported frameworks such as Apache Cordova and IBM Worklight. This framework is managing all UI activities while other frameworks provide the access to native features.
46 Sencha Touch is a JavaScript library for user interfaces with focus on the development of native-acting applications based on HTML5, CSS3 and JavaScript
47 Appcelerator Titanium is a development platform for applications based on web technologies. The platform integrates existing frameworks like jQuery and provides an API for platform-independent access to native UI elements and hardware features.
48 IBM Worklight is an application platform for developing cross-platform mobile applications using standard web technologies

- **Web-based application frameworks** are focused on web applications, web services, and web resources to simplify and unify repetitive development tasks such as the construction of layouts, support of data communications to backend servers, and data manipulation. Known representatives of this category are the Zend Framework[49], Django[50], GWT[51], JSF[52], jQuery[53], and YAML[54].

3.4.2 Framework Architecture

The development of frameworks requires the consideration of different aspects. From one point of view, there must be a clear understanding of the framework objective, so that the corresponding framework type can be applied. This is followed by a classification in various subaspects that the framework is supposed to be covered. Accordingly, there is often a risk during the framework development that people attempt to take too many subclassifications into account which lead to a large number of design patterns, scenarios, programs, and features in the framework. Frameworks must always aim to reflect structures that address specific needs of larger developer circles which are also used frequently. Otherwise, the complexity of the framework and the maintainability would exceed the potential benefits by considering too many concerns. Such bad approaches are called Anti-pattern[55]. This would result in low user acceptance of developers using the framework. Architectural patterns represent pattern types at the highest level of

[49] Zend Framework is an object-oriented framework for the scripting language PHP

[50] Django is an open source web framework for the programming language Python

[51] Google Web Toolkit (GWT) is a toolkit from Google

[52] JavaServer Faces (JSF) is a web framework for developing graphical user interfaces that run within a servlet container such as Apache Tomcat or IBM WebSphere

[53] jQuery is a cross-platform library written in JavaScript, which is used for navigation and manipulation of the Document Object Model (DOM).

[54] Yet Another Multicolumn Layout (YAML) is a framework based on Cascading Style Sheets (CSS) that provide a cross-browser collection of YAML-specific layouts.

[55] Anti-pattern means those aspects of ill-conceived or unnecessary approaches and thus represents the counterpart to other generally valid types of pattern. A number of aspects such as incomplete functions (Smoke and mirrors), cross-domain application (scope creep), a lack of a recognizable software architecture (Big Ball of Mud), or an inadequate distribution of tasks (god object) are considered in this regard.

abstraction by addressing abstract and global problems, while idioms and design patterns focus on a specific part problem. The architectural pattern defines the basic structure of an application and describes interactions between different components of the architecture. Due to the variety of architectural patterns, these patterns are divided according to the relevant aspects into appropriate categories. Subsequently, the well-known patterns with examples are listed for the categories mud to structure, distributed systems, and interactive systems [Cf. Bus00, p. 28].

Mud to structure

- **Pipes and filters architectural pattern** describes the data stream processing of all components of a system. The individual sequential processing steps of a component are described in terms of data manipulation (e.g. create, update, delete) and linking of filters that are connected by channels (pipes), such as Printing and Converter Services (e.g. PDF), data mining [Cf. KH06], and digital camera [Cf. ZG10, p. 125]
- **Layered architecture** assigns essential aspects of a system to different layers. The dependencies of relationships between the layers are clearly defined and restricted. A layer can only interact directly with the higher and lower layer, but not communicate with other layers directly, e.g. OSI reference model and 3-tier software development (Data, Logic and Presentation Layer).
- **Domain-Driven Design (DDD)** follows the modeling approach for complex object-oriented software solutions mainly focus on problem domain[56] and domain logic [Cf. Nil06, p. 469]. The term was formed by Eric Evans through the domain-specific perspective in the design. DDD requires for the creation of a domain model an interdisciplinary understanding of complex, technical, and subject-specific contexts. The core of the domain model, which is based on a variety of concepts focuses on the business logic [Cf. Eva03, p. 75], which assumes a layered architecture, e.g. TYPO3 Flow[57].

[56] In software engineering, a problem domain is a clearly delimited problem field, in which domain-specific applications primarily address this area.
[57] TYPO3 Flow (previously called FLOW3) is a PHP framework for web applications [Cf. TYP13]

Distributed systems

- **Service-Oriented architecture (SOA)** is a pattern for distributed service-oriented architectures to provide (mostly web-based) services. Business processes are depicted in structured form by orchestration lower abstraction levels and thus allowing the design and composition of appropriate services at higher abstraction levels. By encapsulating certain individual tasks, services can be realized in a contextual framework, e.g. Financial Services, web shops (Amazon), search services (Google Search).
- **Peer-to-Peer (P2P)** do no need a central instance in contrast to Client-Server (C/S) pattern, but can distribute tasks and services within the network to all equal participants. Thus, each participant provides functionalities as client and server for other participants. Each user can provide specific services in the network according to the technical qualification (e.g. network bandwidth, hardware), e.g. file sharing (Napster, Gnutella).
- **Client-Server (CS)** pattern describes the classical communication between two nodes in a network with the need for a central server. Clients who want to claim and consume specific services may require such functions from a server. The server in turn offers its resource to other clients in the network as well as collects information and sends a corresponding response back to the client, e.g. e-mail server, FTP server.

Interactive systems

- **Model View Controller (MVC)** pattern is classified according to the respective tasks into three components of the content presentation, namely View, data model (Model), and control of the business logic (Controller). All three components are interrelated and interact through defined interfaces. Due to the flexible division of responsibilities, individual components of the architecture pattern can be exchanged and reused. For example, user interfaces can be exchanged seamlessly without the need for adaptation of the underlying data model and business logic, such as Content Management Systems (CMS) (e.g. Joomla [Cf. Joo13], Yii Framework [Cf. Yii13] and Typo3 Extensions [Cf. Ass12]).

Moreover, there are many other patterns that are not further described in this section, because they are not relevant to the context of this research work. Due to the domain-specific requirements in the PLM environment, the architectural patterns domain-driven design (DDD) is considered for the Mobile Feature Framework and thus deeper investigated.

3.4.3 Framework Design

In addition to the selection of an appropriate architecture pattern, corresponding set of rules must be taken into account during the design and implementation phase of the framework. A rule set consists of various different fundamental principles, which are summarized from a variety of perspectives (also called viewpoints). In this regard, there are several models and approaches for the framework development that assist in the selection of reusable patterns, supports modeling of the framework, strategies for the implementation as well as use of appropriate testing methods. Moreover, frameworks must meet the different requirements in terms of performance, scaling, extensibility as well as adaptability and further requirements. In this regard, the steps listed in Table 3.11 are considered in the framework conception.

3.5 Summary

The previous sections presented the essential aspects of mobile users, mobile devices, mobile applications and PDM systems, mobile PLM solutions as well as described the gaps in the prior art. Moreover, it addressed existing frameworks. As shown in Table 3.12, the requirements described in Chapter 2 cannot completely be covered for the PLM context by any of the examined frameworks. For mobile standalone applications without reference to any PLM subject, frameworks like jQuery Mobile, PhoneGap (Apache Cordova), Sencha Touch, Appcelerator Titanium, and IBM Worklight are suitable. These frameworks allow cross-platform implementations of secure and networked applications without integral PLM approach. The developer obtains a platform provided that focuses on individual mobile applications. A generic and centralized approach

Steps of the framework conception
Determining the scope of the framework
- Type of framework and knowledge validation - Selection, decision-making, and abstract design of scenarios
Modeling
- Design of the layered architecture and class hierarchy - Design of the object model, data model and security model - Interface design and exception management hierarchy definition
Programming
- Class definitions and general coding rules - Naming guidelines and general naming conventions - Global functions/variables and class naming, structs and interfaces - Data access management
Test strategies
- Selection of appropriate test methods - Generation of test scenarios and test execution and logging

Table 3.11: Important steps of the framework conception

in consideration of recurring mobile activities of PLM users does not take place. In addition, PLM concepts are generally not considered by these frameworks. Today's PDM systems offer a variety of native and partially proprietary integrations for CAx software solutions, ERP solutions, and other software systems, but not for mobile contexts and mobile features. Integration modules are of particular significance of the PLM concept, since various business units have different perspectives on the product life cycle and accordingly also use various applications. In order to efficiently provide all information of these applications up-to-date, consistent and centralized by the PDM system for other departments, the PDM system requires individual integrations for data and process management between various systems such as CAD system. For the mobile environment, no corresponding mobile PLM concepts exist to integrate the various phases of the product life cycle in mobile situations and to support them. The PLM vendors do not provide developers specific components for the mobile environment. Only a common API to access the PDM system is provided without any consideration of the mobile context separately. This applies for mobile devices as well as mobile users. Moreover, the developer needs to know the specific and proprietary data model of the PLM vendor to bring all mobile activities into conformity with PLM processes. The developer cannot use existing patterns of interaction with mobile users and PDM systems and must therefore re-implement all logical and visual components for communication, interaction, and persistence of the data. Mobile features are not even considered in the first place by the PLM vendors. In summary it can be stated that mobile features in context of PLM so far neither in research nor in industrial applications are to be found or much less have been integrated.

Properties \ Frameworks	jQuery Mobile	Apache Cordova	Sencha Touch	Appcelerator Titanium	Worklight	ENOVIA API	Teamcenter API
Consideration of Mobile User Behavior							
Capability to consider user behavior	-	+	+	+	+	-	-
Server-side user behavior determination	-	-	-	-	-	-	-
Mobile Feature Support							
Support for native hardware-access	-	+	ø	+	+	-	-
Parallel access of native hardware-features from server-side	-	-	-	-	-	-	-
Server-side and non-interactive control of native hardware-features	-	-	-	-	-	-	-
Support for native mobile feature extension	-	+	-	-	-	-	-
Mobile Feature combination for workflows	-	+	ø	+	+	-	-
Switchover of workflow interaction between web-based and native UI	-	-	+	+	+	-	-
Support of Mobile Devices and Communication Protocols							
Support for multiple mobile platforms	+	+	+	+	+	-	-
Integrated communication channel separation	-	-	-	-	-	-	-
Integrated support for message queueing	-	-	-	-	-	-	-
Integrated support for MQTT	-	-	-	-	-	-	-
Integrated support for WebSockets	-	-	-	-	-	-	-
Product Data Management Support							
Integration capability for PDM systems	-	-	-	-	-	+	+
Access PLM objects	-	-	-	-	-	+	+
Provision of PLM patterns	-	-	-	-	-	-	-
Integrated mobile support for PDM workflows	-	-	-	-	-	-	-

Legend: **+** Supported **ø** Partially Supported **-** Not Supported

Table 3.12: Comparison of framework characteristics

4 Overall Model

This chapter introduces the necessary steps in the model building process of the overall model in context of the Mobile Feature Framework development. Firstly, the objective, approach and methodology are described in Section 4.1. Section 4.2 and 4.3 analyze the context of the subject-specific domain and a common definition of the terminology of different contexts is made. This is followed by the extraction of patterns based on the specification of user stories and detailed scenarios in Section 4.4. The development phases of the subject-specific model as well as the division of the architecture into layers are carried out in Section 4.5 and 4.6. Section 4.7 represents the modeled components and describes their functionalities. Section 4.8 describes the object and service dependencies of the multi-tier architecture. Finally, Section 4.9 covers the cross-component sub-models of the framework architecture.

4.1 Path

This section is intended to formulate the objectives for the following chapters. Therefore, an approach is presented that describes the procedures for the modeling framework. The methodology relies on an established process model of software development.

4.1.1 Objective

In Section 3.3.2 it was shown that current PLM concepts consider mobile features in the product life cycle insufficient and therefore substantial research and development efforts are still needed to provide component-based features as well as enable cost-effectively integration of mobile features into processes and workflows of a system. The result of missing feature components in the PLM field is perceptible by restricted user experiences in the mobile sector. The specific issue in the provision of mobile component-based features is related to the interdependence of mobile features, mobile devices with the appropriate mobile platform as well as mobile users. Accordingly, the specification of mobile features has impacts not only on the realization of the respective mobile platforms but also on cultural and social aspects of mobile users. This requires a close integration of the fields of mobility, users and PLM to take the relevant aspects during the development phases into account (see Figure 1.6).

Procedural models of the agile software development are suitable in this respect in order to develop a Mobile Feature Framework in modular components for PLM concepts efficiently, which also possess the specific abilities to adapt new requirements, new mobile technologies and market changes. Such models assist the framework development through a methodical approach in the respective development phases. The development of a Mobile Feature Framework for mobile PLM users has to support the execution of processes through the integration of mobile features to cope with specific tasks more effectively and efficiently and to enable completely new and innovative application scenarios. The framework to be developed should take into account patterns that are identified, analyzed and formulated generically in the modeling phase in order to apply them in the implementation phase after modeling of mobile business processes. This is necessary to ensure that these patterns are reused in a variety of applications for mobile PLM applications.

4.1.2 Approach

Component-based design of the framework and mobile features

PLM vendors develop individual customer-specific software solutions for service projects as well as provide appropriate OOTB solutions (see Section 3.3.2). During the development of such software solutions, corresponding development methods and tools are used mainly, but customer requirements are not always met sufficiently. The reasons are usually multi-faceted and complex. These circumstances sometimes lead to delayed completion of the custom software solution caused by a lack of industry knowledge[1], exceeded development costs, insufficient software quality and above-average costs for the continuous operation of the software[2]. In computer science, the approach of top-down and bot- tom-up design is used to get a better understanding of software systems and processes. Therefore, the top-down design point of view is taken from an abstract perspective on the system. Initially, all details of the system do not play a role until the system is broken down into smaller components, which then include precise definitions and detailed specifications of the components. This approach is used in other application fields such as the design of mechatronic products in the automotive industry, the style of leadership in management as well as sales strategy in marketing. For the ICT sector, the re-use of quality-assured components is essential to ensure low development costs, the demand of highest quality and a very short *Time-to-Market* (TTM). Therefore, the component-based design of the framework has already a major importance in the modeling phase. The advantages of the feature framework and mobile applications are that the structural design of the framework provides the basis for further extensions and facilitating maintenance. Moreover, various mobile features can be designed as components of the framework in parallel and independently from each other, thus resulting in cost and time benefits. The reuse of feature components of the framework for mobile PLM applications also increases the quality of individual components, because identified software errors can be detected and solved effectively and efficiently through the multiple uses of components. Thus, it is no longer necessary to implement and maintain mobile features for each mobile PLM application separately.

[1] PLM solutions are used by a variety of industries that require an appropriate knowledge of the industry to understand industry- and customer-specific processes. This knowledge is essential for the development of software modules.

[2] Increased operating costs of the software solution arise, inter alia, through the risk of high costs of introducing new software release containing customer-specific adaptations. These adjustments must be individual validated and modified for each new software release of the PLM manufacturer by the customer.

This in fact reduces the operational cost and required changes need to be applied only in one place (framework component). Consequently, the use of framework and feature components leads to quality, time, and cost benefits in the software development of mobile PLM applications.

Specification of a demand-fulfilling Mobile Feature Framework for mobile PLM applications

While monolithic applications (e.g. DOS and Windows 3.x applications) prefer a tight coupling of functionality, a paradigm shift to multi-layered and service-oriented architectures has taken place over the past few years. Monolithic applications are based on homogeneous structures, whereas multilayer architectures respect a strict separation of functionalities by corresponding layers. For this purpose, a separation is made for the presentation layer, business logic, data management, provided services, and other layers. Over the last years, the architectural pattern SOA has prevailed, which causes a rethinking of the architecture construction. The paradigm shift led to the consideration of services from the different levels of abstraction. Frameworks spend their attention to multi-tier architectures in order to avoid monolithic structures and to ensure the efficiency (e.g. scaling, integration, adaptability). Therefore, functionalities are generally managed and provided separately. In addition, all layers support a separate provision of basic functionalities such as logging, exception handling, and security to avoid multiple implementations in different layers. However, the core of framework provides reusable and coherent structures for mobile applications which are based on appropriate design patterns. Moreover, existing software systems are integrated through adapters and integration modules. The fact that generic pattern is reusable instead of the implementation of individual software solutions, the costs and time of the application development are reduced while the software quality is increased. The composition with the corresponding relations of components is described by architectural models to reflect the respective design patterns by corresponding components of the framework. The architecture style defines the scope of the framework and the place of the perspective for objects, information, systems, networks, and security. Whereas the focus of the *Software Architecture* is on the software development with the corresponding software components, the *System Architecture* emphasizes the interaction of components between different systems consisting of hardware and software architecture as well as related interfaces.

4.1.3 Methodology

In order to ensure that the structures with the appropriate relationships and the layered architecture of the framework can be designed to a model, the use of a procedural model of the agile software development is beneficial. The process model *Feature Driven Development* (FDD) is selected as an iterative, incremental and agile approach, because the focus of the development phases is on component-based features that appear to be suitable in context of the development of Mobile Feature Framework. FDD has a waterfall-like structure in which individual processes are sequentially arranged and the switch between process steps is not envisaged. For this purpose, the FDD defines five primary processes, which have to be executed (see Figure 4.1). In context of this work, the provided role model of FDD is not applied. The first process defines the development of a subject-specific model of the corresponding domain. In context of this work, Domain Driven Design (DDD) is used as concept for the domain modeling, because the relevant business and professional relationships of PLM are placed at the center of modeling.

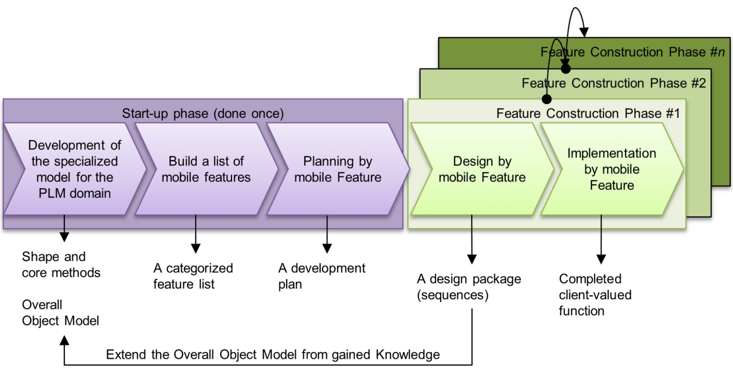


Figure 4.1: Feature Driven Design (FDD) process (adapted from [Cf. Amb12])

The core of DDD represents the domain model and the common used domain-specific language. In so doing, the domain model captures the types of objects, structures and processes of the specific PLM domain with object-oriented concepts, which are explained in Section 4.5. In the following the five primary processes are described:

1. Development of the subject-specific model of the PLM domain

Initially, Section 4.2 defines the conception of a subject-specific model for the application domain using DDD, which also identifies the corresponding contexts. This is followed by the definition of the ubiquitous language for this application domain in Section 4.3. The concept of the subject-specific model of the application domain PLM is covered in Section 4.5 by prior consideration of the extracted design pattern in Section 4.4. The objective of the first process phase is to identify and consider an overview and common understanding of the specific domain. Therefore, it is necessary to determine the individual components of the overall system through the generation of sub-models for the corresponding system areas, which have to be created and integrated to a common subject-specific model. This will provide an overview of the system scope and responsibilities of framework components. The models created are transferred to a UML tool[3].

2. Creation of a mobile feature list

The second process phase is based on the modeled PLM domain of the first process phase for the creation of appropriate feature lists. Therefore, the feature lists are derived from the extracted pattern of the user stories which are described in Section 4.4. For this purpose, each feature list represents a business activity (see Figure 4.2). The domain model is splitted into smaller sections (so called *sub areas*). These components

[3] Unified Modeling Language (UML) is a modeling language in software engineering and provides a uniform notation in order to describe the design of software components and systems in a visual form.

consist of corresponding business activities (*Activities*), which include individual business steps (*Activity steps*). A business activity reflects a feature list, which is grouped by several features. Thereby, the business steps of a business activity define the category, scope, and form of a feature list. In this case, a feature represents a function. A feature is defined by the scheme ‹*Action*›‹*Result*›‹*Object*› [Cf. Luc12]. The results represent structured feature lists which describe clear, concise and uniform requirements by using the domain model.

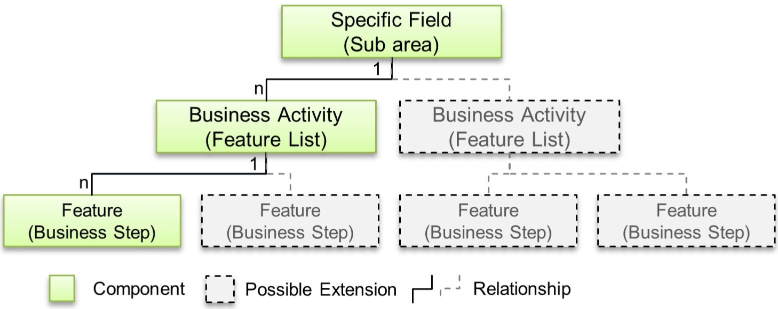


Figure 4.2: Feature Driven Design (FDD) hierarchy principle (adapted from [Cf. RW08])

The feature lists consist solely of mobile (device-specific) features and do not contain system-specific features as is usual in FDD. System-specific features are considered in Phase 1 because the outcome is a reference implementation of the model and does not aim a productive overall system for professional customers. The features of each business activity contain technical features (e.g. Logic: Create a list) and artificial features (e.g. User Interface: Show list) to create a better understanding of relationships between features and the business activity. In case that a feature is too costly, too time-consuming, or too complex, it will be divided into several individual features.

3. Mobile feature planning

Once the feature lists are created in the second process phase, the third process phase is performing the planning of the features. For this purpose, a development plan is created that specifies the order of the features in which they should be designed. In this case, the dependencies and the respective complexity of the features are crucial.

4. Mobile feature design

In the first phase, the domain model not only has been developed to obtain an illustrative overview but also has described all objects and relations to each other as well as services and components. However, classes and methods have not been described in detail and mobile devices features have not been modeled. These activities are the responsibility of the fourth process phase that creates corresponding sequence diagrams of features in order to be able to refine the class models. The sequence diagrams have the advantage to determine the methods in detail and classes which are necessary for the feature implementation. Afterwards, the class and method prologues are generated. Additional documents describe specifications of third-party systems and applications. Finally, the results are validated and inconsistencies are removed. In case of technical issues, solutions are developed in cooperation with subject matter experts. This process is known as a *Design Inspection*.

5. Mobile feature construction

The last phase of the process includes the implementation of the features. To ensure the software quality, the implemented components are reviewed using suitable test methods and appropriate code inspections are performed. The code inspection includes the verification of the properly applied usage of specified guidelines for subjects, such as application of the ubiquitous PLM language in the programming code, usage of algorithms as well as avoidance of *Anti-pattern*. Finally, the implemented feature is supplied to the software build process, which produces as result an automatically generated software release.

4.2 Analysis of the Framework Context

Due to the interaction with PDM systems, respectively different contexts exist for stationary and mobile users. According to the terminology of DDD, the term *Bounded Context* defines an enclosed model or subject referred by users. For this purpose, each identified sub-domain has its own context and isolated the corresponding know-how. Thereby, a context has no knowledge about the operations of other contexts. Each context includes different elements such as *Entities*, *Value Objects* and *Aggregates*. Entities are defined by their identity and status, whereas Value Objects focus on the properties and attributes. Aggregates combine Entities and Value Objects as well as their relationship to each other into a single element. PDM systems and mobile environments have their own context, which requires appropriate interactions due to the integration of mobile features (e.g. message transmissions, file transfers, feature calls). The components that are not part of the domain model and thereby cannot be represented in a generic model are described in *Generic Subdomains*, such as reporting and statistics. By the fact that an interaction between multiple contexts is mandatory, a translation between the concepts of the mobile feature domain is required. The contexts in DDD are of particular importance, because they have an influence on the collaboration of objects and environment-specific properties, such as business rules. This means that objects in the PDM context have to follow another set of rules as objects in context of mobile features. For example, different access permissions to specific objects are defined depending on the life cycle status. Once the life cycle status of the object changes in the PDM context (e.g. through promote or demote operation on an object) and the object acts outside of the PDM context (e.g. mobile context), the access privileges to the object maintain the same permissions. Therefore, such PLM specific regulations of the business logic should be included in the domain layer.

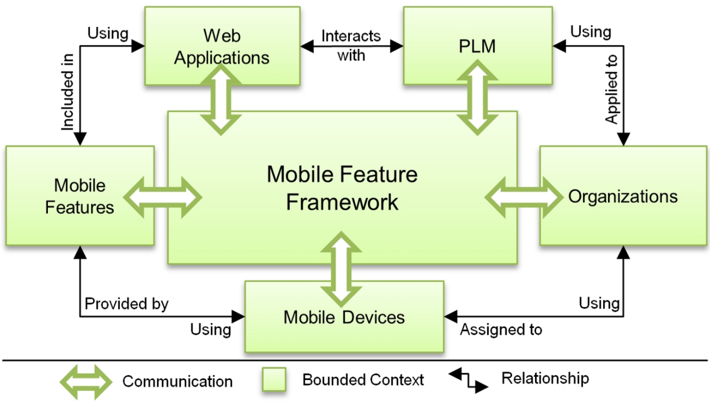


Figure 4.3: Context map

This section covers the limits of the framework by describing the contexts and defining their respective validity range. Thereby, this should ensure that there are no overlapping contexts and communications between the contexts are carried out exclusively by using defined interfaces. The *Context Map* (see Figure 4.3) is used to obtain an overview of all contexts that are relevant in the surrounding of the Mobile Feature Framework. In context of the framework following bounded contexts exist:

- The context of the *Web application* deals with the workflow components and processes of PDM systems. In that regard, the user experience on mobile devices is important. Mobile web applications have to be seamlessly integrated into existing PLM processes and also have to support the integration of mobile features into mobile workflows.
- The *Mobile Devices* context places the perspectives on different types of devices to take into account the relevant aspects of mobile platforms for mobile features. The context boundary runs through the context of Mobile Feature.
- The context *Mobile Features* considers the interactions with mobile users in a workflow as well as handling with specific information. The context defines its boundaries at the Web application in which mobile features are embedded in a workflow and called from a

mobile web application. A mobile feature can contain not only platform-independent software components but also a combination of device and platform-specific components. The device and platform-specific components represent the boundary to the context *Mobile Devices*.

- The context of *PDM* represents the corresponding component from the respective PLM concept. Since PDM systems have already their own internal models and contexts, they cannot be mixed and combined with the framework caused by framework requirements. Therefore, PDM is treated as a separate context. Because the framework focuses on the integration of mobile features in context of PLM, other possible integration modules such as ERP are not considered in this work.
- The organization as a context reflects the people and business units, which are often mapped in the PLM context. It must be noted that each context has its own Ubiquitous Language which has to be considered accordingly.

4.3 Ubiquitous PLM Language

The most important task in the domain modeling is the unified and common language of the requirement definition which covers the understanding of the various subject-specific languages of their proper context and meets the expectations of the stakeholders, such as designer engineers, developers, reviewers, project leader, and service people. Normally, the requirements for a system are described in the form of a set of user stories and use cases. The described application scenarios are usually formulated inaccurate on an abstract level by users. Therefore, it is important in the design phase to understand the specific context (e.g. industry-specific PLM workflows or processes) in order to take into account the appropriate design for the modeling and implementation. The required knowledge can be delivered by people who have already extensive experience in the domain. In this regard, the experienced people have to fill out appropriate requirement documents capturing the respective perspectives in order to document the expected functionality that provides the business benefits. In addition, direct conversations with the domain experts by asking questions can help to understand the specific terminology and to reflect the terms in the code, because they daily use the industry-specific vocabulary. PLM terms such as "Lifecycle", "Process" and "Model" should be accordingly reflected in the framework. Thereby, the gap between the business world in the PLM field and the technical terminology is minimized and an additional benefit is generated.

Commonly used language must be specified as a base (DDD calls this language as Ubiquitous Language) in order to get a common understanding between people from the PLM world, developers of web applications as well as mobile devices and platforms. For this purpose, a description of the PLM professionalism, the elements of the subject-specific model as well as classes and methods are followed. A dictionary is used to document the ubiquitous terms. This requires a deeper examination of PLM processes to identify specific PLM terms and include these terms in the dictionary. The created dictionary for the PLM domain with the corresponding definition can be found in Appendix C.3.

4.4 Extraction of Patterns from User Stories

PLM concepts cover a variety of application scenarios differing from the respective industry and individual customer requirements. Customers describe abstractly formulated user stories about what functions and features the software is supposed to provide. Typically, user stories are formulated in the format *‹Role›‹Feature›‹Benefit›*. For example: *The engineer wants to perform a database search by his voice, so that the search term must not be entered via keyboard of the mobile device.*

The user story is a suitable starting point, but it is in the current state inappropriate for the domain modeling. Therefore, the user story must be described in an adequate level of detail and derived accordingly to application cases for the domain modeling. Based on the use cases (see Figure 4.4), the appropriate requirements are derived which has been described in Chapter 2. The classic "Type a phrase" scenario already exists for all conventional PLM concepts. Thereby, a phrase is entered and submitted into a defined form mask using the keyboard. The server-side component receives and analyzes the corresponding phrase, determines the result and sends it back to user's mobile device. Depending on the complexity and length of the phrase as well as the mobile user's ability, the duration of the input operation using the keyboard can vary enormously. The previously mentioned scenario represents the basis of the contribution for the majority of mobile PLM solutions, which are used by mobile users to contribute using the conventional way. However, mobile features enable a number of new ways to consume and contribute information through mobile devices, which are described precisely in this section based on the fundament of user stories.

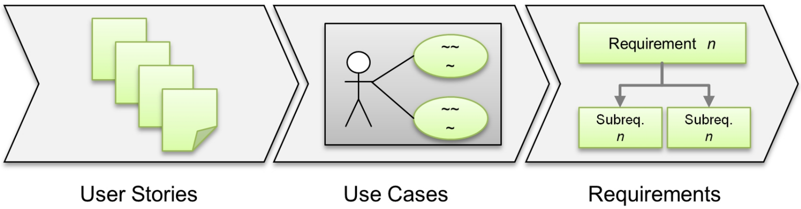


Figure 4.4: Extraction path

From the variety of application scenarios, eight primary groups were derived and abstracted. These abstracted application scenarios cover most of typical scenarios from the domain of classical and stationary PLM concepts and can contribute added value to the mobile PLM sector. The groups are composed mainly by activities (verbs) that express an action or a state of an object. Accordingly, objects can be observed, identified, analyzed, located, examined as well as objects can collaborate and information over objects can be contributed. However, the grouping can also be organized from the feature perspective and therefore a distinction into three groups takes place: a feature interacts with the user independent from foreign components, a feature works within a dependent component (e.g. workflow) as well as a feature works independently and automatically from other components without user interactions. The classification by object activity appears to be more suitable. As a result, the groups will be outlined only briefly, but they will not be further pursued and applied in this work.

- **User-driven mobile features:** Mobile features that perform interactions with mobile users and are not linked to a workflow represent the first feature group. These are a completely new set of application scenarios, which actually does not exist in classical and stationary PLM concepts. Nowadays, information is retrieved from stationary PDM systems using classical input devices like mouse and keyboard. This newly created group extends the capabilities in which information can be retrieved with the support of mobile features. For this purpose, the respective mobile features are used to perform the interaction with the users to interpret the results and to perform a defined action.
- **Integrative mobile features:** Mobile features that are used in the context of workflows or processes represent the second group of application scenarios. This is always the case when the user acts within a workflow while mobile features support the various workflow steps.
- **Device driven mobile features:** This group includes scenarios in which external events are transmitted to the mobile device. These events can trigger appropriate actions on the mobile device and affect its behavior.

Section 4.4.1 presents and clusters the individual groups and application scenarios from the user perspective. A use case is created for each application scenario that reflects the user story. A pattern describes only a part of the cluster without defining the concept and the detailed implementation. Therefore, a pattern can addresses various contexts and thus various mobile features can be generated and varied greatly in design and implementation.

4.4.1 Discover Objects

One of the use case clusters is represented by the examination of objects for mobile features. Objects are only examined when an activity of the mobile user or mobile device is performed and this causes a state change.

In this case, the examined object is not primarily target, but the observer of the object who registers to a state change based on events and initiates the examination action of the object. An observer can register the shaking of the mobile device, recognize gestures and languages as well as perceive changes of context-specific properties.

The observer is informed by the object about status changes in which the object can exist in the real or virtual world. The observed item provides the functionality to communicate state changes to the observer. For this purpose, the observer must be registered to the object. The registration is performed to the object using standardized interfaces. This design pattern from the software development corresponds to the well-known *observer pattern* of *Gang of Four*. Based on this generally applicable pattern, the following Observe-patterns of this cluster for the PLM context has been derived for the Mobile Feature Framework.

Discover Things by Shaking the Device

In this application case, appropriate action calls are executed by shake sequences. The shake sequence is defined in advance and triggered by the mobile user. In this regard, the shake sequence can be specified by the number of shakes, speed and orientation of the mobile device to allow different action triggers. The underlying action can trigger either a direct function call to display target information (e.g. list of all project activities) or initiate an intermediary function for further interaction with the mobile user. Thereby, the mobile device would provide to the user for instance a voice-based search functionality that calls the native language interface through a shake sequence and the interpreted phrase would initiate a corresponding function call. This represents a composition of mobile features by combination of mobile features, which leads to completely new applications. The definition of voice-action commands is freely configurable, so that native as well as web-based actions can be performed. The use case is described in Appendix D.1.1.

Discover Things by Gesture Recognizer

The built-in mobile feature for gesture recognition allows non-contact interactions with the mobile device through hand movements above the display. It enables the examination of 3D objects in a completely new way, without the necessity of a computer-based data glove[4]. In addition, gestures can be used for controlling and navigation through an intelligent combination of the interaction between the mobile user and the user interface in applications. Thus, gestures can be used to enable an individual encoding of a function call. For example, a vertical movement of the user's hand would call the list of pending activities while a horizontal hand movement would present user's project appointments of the day. The use case is described in Appendix D.1.2.

Discover Things by Speech Recognition

The speech recognition scenario is a type of query language for information and interaction execution, which initiates underlying actions for a mobile user by corresponding voice syntax. Through naturally spoken language, personally defined functions can be called while the voice data is recognized and interpreted back to the feature by voice recognition software. The scenario covers three language options:

1. Simple phrases for general inquiries, for example *"My Workspace"*
2. Specific requests, for example *"Show 3D-Model xy"*
3. Interrogative sentences, for example *"How many tasks are pending for today?"*

In order to ensure this functionality, extensive logical structures must be implemented on the server side. The corresponding use case is described in Appendix D.1.3.

[4] The data glove is used as an input device in virtual reality for navigation and orientation in virtual spaces.

Discover Things by Facial Expression

In some mobile situations it may happen that the social environment does not allow acoustic or gesture interactions between users and their mobile device. For example, noisy environments predominate over the mobile device, the mobile user's hands are already blocked with objects or the person has physical disabilities. In such situations, the control of simple activities by expression should take place. The mimic is understood not only as the visible movements of human facial expressions but also as the movements of the facial muscles form different facets. This allows the camera of the mobile device to interpret different movements. Table 4.1 shows a possible association between facial expressions and action defined in the PDM system. Simple interactions can thus be carried out by different facets. However, more complex interactions are complicated to turn into reality and must be combined with combinatorial movements and sequences (e.g. one wink corresponds to "*Yes*", two winks correspond to "*No*"). Cultural and technical aspects represent additional challenges. Thus, the facial expressions strongly differ between European countries and Asian countries. The use case is described in Appendix D.1.4.

Facial expressions	Interpretation	Defined Action
Frown	Thoughtfulness, rejection	Release of objects is rejected
Wink	Affection, approval	Release of objects is approved
Tense facial features	Social rejection	Collaboration or communication is rejected
Eyelids slightly raise	Interest [Cf. Eil13, p. 100]	Show additional information about the object

Table 4.1: Mapping facial expressions and PDM defined action

Discover Things by User Context Consideration

This scenario considers a subarea of the user context by the integration of context-related factors in the mobile web application during the interaction with the user. This also includes visual influences (luminous intensity), climatic changes (temperature) and acoustical events (acoustic). The scenario covers three sub-scenarios and is described in Appendix D.1.5

- Visual Influences: Adapting the user interface, for example *Day-night mode*
- Climate Changes: Location changes affect the application, for example by *displaying certain test data at low temperatures*
- Acoustic Events: The application logic and the kind of information presentation are influenced based on acoustic events in the environment of the mobile user. For example, persons in the production hall would preferably present visual information (e.g. AR prepared information) instead of acoustic information (by reading texts aloud for visually handicapped persons). The application logic would be primarily influenced by the mobile acoustic feature.

Observe User Behavior to Avoid Interruptions

Muting the mobile device can be interpreted as a natural behavior when a mobile user does not want to be disturbed. The volume control also allows conclusions to be made about the actual user context. Thus, a low-adjusted call volume may imply that the mobile user indeed can be disturbed, but their immediate environment should not be disturbed. However, a maximum volume could imply that the mobile user is surrounded by an environment with an excessively high volume level. When a mobile feature (e.g. microphone) cannot be used, alternative methods such as the recognition of the user behavior could support to determine the actual context information. The use case is described in Appendix D.1.6.

4.4.2 Identify Objects

The identification of objects of the real world represents another feature group and can be carried out by various ways. In order to enable object interactions, objects must be previously identified for their interactions by the mobile user or a software-based algorithm. Since the mobile user does not have the necessary background knowledge, the identification of objects usually takes place using software-based methods based on defined characteristics. The captured features are subsequently analyzed and evaluated in order to determine the identity of the object or to present a limited selection of possible objects to the user. Afterwards, the user has to determine and confirm the identity of the object through additional characteristics. The software-based identification can be carried out by visual, auditory, olfactory[5] as well as gustatory[6] and tactile perceptions[7]. The visual and auditory perceptions are already being used in software development in order to identify objects by unique codes, image recognition, and acoustic snippets. The olfactory perception as an information channel is still in its early days of research, since the processing of smelling has a high complexity and practical applications so far hardly exist for the mass market. In present research, olfactory systems of living organisms are examined with the objective to be able to transmit the knowledge to a computerized system. Although there are sensors for the identification of gases, but such sensors are limited to a single application purpose of a specific area. The gustatory perception stimulates the sense of taste by chemical stimuli. In the field of Mobile ICT, this perception does not matter, because of lacking technological capabilities. However, this kind of perception is interesting for the identification of bitter substances as well as acidic and toxic substances. For example, the quality of mineral debris and lubricants could be checked. The tactile perception refers to the ability to perceive mechanical pressures. This can be effected by active palpation of objects or by passively actions of the object itself. The research field of robotics is concerned with the subject of object interaction in the physical world

[5] The olfactory perception refers to the sense of smell
[6] The gustatory perception refers to the sense of taste
[7] The tactile perception refers to the sense of touch to perceive mechanical impacts

based on sensor technology. In practice, visual and auditory perceptions are used as primary methods for identifying objects, because all other mentioned perceptions have not yet been sufficiently explored in order to take adequate account for the Mobile Feature Framework in context of PLM. Based on the different perceptions, the following patterns of the cluster have been derived for the Mobile Feature Framework.

Identify Objects by Image Recognition

In this scenario, the camera of the mobile device is used to recognize real objects on the basis of contours and to present the corresponding stored information. In this context, a real object is understood as a body instead of an identification code, which has been attached to the body. The body can be linked to additional information. Therefore, the reality is supplemented by the overlay with graphical information from the PLM backbone. This scenario is a composition of Augmented Reality (AR) and the image recognition and processing fields. The use case is described in Appendix D.1.7.

Identify Objects by ID-Code Scan of Objects

The detection of the identification codes using the camera is described in this scenario. Therefore, a code is captured with the camera, which has been previously attached to a real object. The captured picture is analyzed by the background system which also interprets the ID Code and returns a unique string as a valid result. This string identifies the object to present additional information to the mobile user. In additional functions such as control, administration, monitoring, and maintenance of the object can be made available to the mobile user as soon as the object has been identified. The subsequent possibilities to combine this use case with further complex scenarios vary enormously and depend on the respective industry. The use case is described in Appendix D.1.8.

Identify Objects by Acoustics

The identification of objects by acoustic sequences is primarily used when identification by code or images is not possible. Objects that do not have an ID-code were scratched or attached at sites which are hard to reach (e.g. serial number is attached on the back to the wall) or were installed in a cordoned-off security zone (e.g. welding robot cubicles) can be identified only with difficulty. Object recognition using a camera is not always possible, since other objects block the view to the target object or the light intensity is insufficient. The identification of objects by individual sounds represents a further application case. For this purpose, the objects must generate acoustic signals. For example, conventional combustion engines generate various engine noises depending on the add-on parts. The radiated sound of the engine surface and other attachments (e.g. intake and exhaust systems, auxiliary power units) affect the engine noise in different ways and to varying degrees. While the combustion noise of the diesel engine represents the dominant noise source, the mechanical noise of the gasoline engine characterizes the sound field of the motor [Cf. Fin01, pp. 5–7]. The Machine Acoustics as a branch of technical acoustics distinguishes between “direct” and “indirect” produced airborne sound [Cf. Ang06, p. 3]. Thereby, the sound of an object is recorded (e.g. hum of an electric motor), the captured sequence is analyzed, and the corresponding object is searched in a server-side database. The generic application case is described in Appendix D.1.9.

4.4.3 Explore Objects

The study of objects does not have the intention to identify objects, but has the intention to explore their characteristics such as volume, dimension, acceleration, light intensity, color and condition as well as aggregation state of objects. This allows the user to distinguish and compare differences between them as well as perform object classification. Likewise changes in objects behavior can be recognized and captured. For example, a person with color-blindness could perform the proper wiring of components through object characteristics if the identified color of a cable is also displayed as text to the user. Based on the various object characteristics, the following patterns of the cluster are derived for the Mobile Feature Framework in context of PLM.

Explore Objects by Surfaces

The camera of a mobile device can be used for varied purposes. The detection of surface patterns and textures of an object can help to identify objects and to retrieve object-related information. For example, the camera can be used to perform a color measurement of a clothing material to determine in further steps the actual market price for an entirely new developing collection. In addition, a digital shade chart helps to compare corresponding color pattern to find color deviation from digital shade of product sample as well as support the color selection decision process. For a deeper analysis of a color, a portable spectrophotometer can be connected to the mobile device to determine the thickness of the ink layer and to calculate color differences. Such a mobile feature supports the quality control in a consistent production monitoring and incoming goods inspection of a company, because the measurement ensures more continuous accuracy, which would not be possible even with trained employees. The use case is described in Appendix D.1.10.

Explore Objects by Measuring Characteristics

The fact that mobile devices deliver a variety of sensors means that can be used for examining and measuring objects. Thus, for instance, mobile features supporting the calculation of the distance between bodies, the measurement of the temperature and humidity of body's environment using thermometer and hydrometer, the measurement of the air pressure using a barometer as well as the measurement of the light intensity using light sensors to capture information that is important for the storage of objects. The magnetic sensor of the mobile device allows the rudimentary detection of metallic bodies. As soon as the device gets close to a metallic body, corresponding defined actions can be triggered. For example, to check whether an unknown body contains metal parts or has special magnetic characteristics. Another scenario could display additional information about material components of a body as soon as the mobile device detects and identifies a magnetized object. Additional information may also be displayed for gaseous and liquid substances when the mobile device is connected with a corresponding technical measuring instrument. The generic application case is described in Appendix D.1.11.

4.4.4 Locate Objects

Before an object is identified and properties are examined, it must firstly be localized. The localization of objects can be accomplished by an accurate position determination or by responsive actions when calling object's senses. The claim of object's senses also means to attract the attention of the observer in order to distinguish an object from other objects in the surrounding. The attention of the observer can be achieved by selective response of visual and auditory perception. For example, flashing LEDs and audible sequences of the object. Based on various localization methods, the following patterns of the cluster are derived for the Mobile Feature Framework in context of the PLM context.

Locate Objects by Coordinates

Meetings, events and other appointments might be scheduled at locations where cannot be immediately found by visitors and guests, unless they visit the place for the first time. The location-based feature of the mobile device supports mobile users in orientation and navigation in premises as well as determines the shortest route. The server-hosted appointments are synchronized with the mobile device, so that these data are accessible without the necessity of a permanent connection. The location of a mobile user cannot always be determined using GPS. This is always the case, when users are inside closed premises. However, if vibrations are noticed by the accelerometer of the mobile device, it might be a location change of a mobile user. The combination of different mobile features (e.g. GPS, WLAN and accelerometer) helps to determine the user context accurately. The use case is described in Appendix D.1.12.

Locate Objects by Acoustics

The localization of objects by coordinates represents the most accurate method if the current GPS position can be determined. However, if the user in the basement of a building, the signal is too weak and the receiver cannot determine the GPS position. Different object characteristics must be used for position determination to ensure that despite the object position can be determined. Mechanical components such as motors noises would allow the localization of the object position. Real objects which do not cause earmarked noises (e.g. earmarked noise such as dishwasher and vacuum cleaner as well as not earmarked noise such as smoke detector and smartphone) could use the object control in order to generate or play certain (synthetic) noises for the time period of position determination. The object control could be used to start an object activity producing noises or the playback of sound sequences using the object speaker. The generic application case is described in Appendix D.1.13.

4.4.5 Examine Objects

The examination of objects and their surrounding environment is using additional information layers in order to provide a broader perspective for the user. The user's attention should not be attained solely through text-based information. Instead, the presentation of condensed information takes place in visualized layers, which as a result are easier to grasp and understand. Due to various methods for the layer-based presentation of information, the following patterns are derived for the Mobile Feature Framework in context of PLM.

Examine Objects by Augmented Reality

Native Augmented Reality applications can increase the transparency of past and future activities with visual information by overlapping appropriate information layers on the real environment. From the perspective of a mobile user, the user would recognize which assigned activities have been fulfilled by other persons. An object could be extended with such an information layer to visualize additional technical characteristics in an AR environment. The focus of this scenario is set to the creation of a personal perception by cross-faded visual layers within the AR environment. However, the perspective of the mobile user can be changed as needed to a single object. A tour through a production with a mobile device would create a whole new perception for mobile users by linking a variety of objects (e.g. people, machines, products) with corresponding information from the PLM backbone. The use case is described in Appendix D.1.14.

Examine Objects by Visualization

Visualization components (e.g. Map Services, Chart Tools, Augmented Reality) support the mobile user in the information interpretation through a visual representation. The enormous amount of information can be limited by visual filter to increase the clarity considerably. Thus, only information from the environment is presented to users who have a business interest (e.g. location-based information, group-related information). Visual filters such as the zoom into a map help to reduce the amount of data until the user reaches an acceptable degree of information density. Likewise, additional detailed information of objects should be presented for a partial section of a dataset when the number of objects has been reduced. This means that many objects show few properties, while few objects show more detailed properties to the user. The focus is on the creation of a personal perception by crossfaded visual layers within the visualization. The use case is described in Appendix D.1.15.

Overlay Realities to Simulate Part Assembly

Realistic assembly simulations already exist since several years for digital test models, which are referred to as Digital Mock-Up (DMU). The counterpart to this is called Physical Mock-Up (PMU). In this case, physical models are used for assembly simulations. The combination of mobile features (such as camera, sensors) with the corresponding PDM data allows such a simulation with physical and digital components.

The merging of the concepts DMU and PMU in this work is referred to as Augmented Reality Mock-Up (ARMU). The use case is described in Appendix D.1.16.

Overlay Realities to Discover Facilities

The simulation of digital production lines, which are necessary to implement the digital factory[8] provides elementary components for hedging the production planning. Present simulations rely on the full visualization without the integration of already existing physical production objects, such as plant control units and machinery. The linking of physical objects of a production line with PLM information would enable a broader user perspective to provide additional visualized runtime information during production. The use case is described in Appendix D.1.17.

Overlay Objects for BOM List Synchronization

The visualization of bills of materials (BOM) for real components within an augmented reality can be used to get a better understanding in the assembly. A field of application is especially the manual assembly as well as service and maintenance in the industry and service sector. For this purpose, existing physical components are compared with a virtual overlapping parts list using mobile features (AR using camera and sensors), so that missing components can be detected immediately by the mobile user. In addition, the mobile user is assisted in the execution of tasks by visually overlaid steps (e.g. step by step instructions to replace parts). The use case is described in Appendix D.1.18.

[8] The term Digital Factory serves as a "generic term for a comprehensive network of digital models and methods including the simulation and 3D visualization. Their purpose is the holistic planning, realization and continuous improvement of all essential factory processes and -resources in conjunction with the product." [SW04, p. 400]

4.4.6 Collaborate with Objects

The interaction and collaboration between objects represent another feature group. In this case, objects are in close dialogue with other objects to achieve a common objective. Thereby, objects can be represented by drawings, people, communication tools, remote diagnostics tools, and other object types. Objects can influence other objects by creating, scanning, editing, deleting, searching, and executing things. For example, a collaboration tool can manipulate a drawing object using appropriate paint functions (e.g. set marking, add additional information) as well as a remote diagnostic tool can control a mobile device (e.g. activate functions, device restart). Based on various possibilities for object interactions, the following patterns of the cluster are derived for the Mobile Feature Framework in context of PLM.

Collaborate with Objects using Online Collaboration Tools

Once a mobile user wants to perform a synchronous collaboration with other persons to communicate interactively through a selected visualized context, collaboration tools make a major contribution. The collaboration tools of mobile devices are designed to provide a virtual collaboration space for distributed people (e.g. Cisco WebEx Meetings and Citrix GoToMeeting). For this purpose, an integration of existing instant messaging tools is considered under the aspect of communications between mobile and stationary users who are involved in the product life cycle. The instant messaging tools have to ensure the synchronous communication between participants of a team under consideration of multi-media communication capabilities (e.g. video and text-based chat). The use case is described in Appendix D.1.19.

Remote Diagnostics and Device Control

External events that are sent over the framework to mobile devices have the primary objective to perform a defined action to control a device. Such actions are related to the context of the device by changing configuration or run-time parameters. In addition, events can initiate appropriate requests for the interaction between the mobile user and the device (e.g. Notification Message). This may happen if server-side events occur for which a subscription of the mobile user exists (e.g. new contribution to the discussion, new or updated document content). The events can also trigger system-related actions such as blocking, control, and shutdown of the device. The type of the event depends on the associated actions. The use case is described in Appendix D.1.20.

Authenticate Applications by Biometrical Characteristics

The camera of the mobile device can be used as authentication method to detect the user's face as well as perform the login process with the PDM system. Alternatively, the fingerprint of the user can be used to perform the login into the PDM system. Both authentication methods require the storage of personalized biometrical information of each mobile user in a directory service of the company (e.g. Fingerprint and face profile). The use case is described in Appendix D.1.21.

4.4.7 Contribute Object Content

Objects from the real world provide continuous information, even if they are offline. The information needs to be recorded and transported only through the use of a medium as well as has to be associated to a digital object. Several media types are available for the capturing of information. Either the user uses medial tools in order to capture information directly from the object or the user provides the information itself by voice or own drawings and sketches. In both cases, the collected information has to be associated with a unique object to be able to interpret the information at a later point in time in the corresponding context. The complement of captured content with additional context-specific data (e.g. time and place) would assist the user in the interpretation of information. Based on various ways of information capturing, the following patterns of the cluster have been derived for the Mobile Feature Framework.

Contribute Content by self-recorded Media and Drawings

Nowadays, workflows use text-based input fields to release data and documents, which capture relevant comments and other messages from users. The user provides short phrases or sometimes firm-specific abbreviations in the comment field, which seem to be not fully understandable for everyone, especially when it involves complex matters. Constructive discussions about design templates and constructions are opened and conducted to any subject in the PDM system. Thereby, discussion threads can be opened to almost any topic related to issues, improvement and administrative tasks. However, access to discussion threads is limited for mobile users. In order to participate in discussions, mobile users have to login into the traditional web-based user interfaces, which are mostly not optimized for mobile use. In addition, media content is usually not sufficiently taken into account and can mostly only be published using file attachments. Visual and acoustic communication components can help to communicate complex subjects vividly and to sustainably promote the collaboration and social interaction within the team. In this context, the interactive and asynchronous communication is focused. Mobile features (Video and audio recording, capturing photos, share the actual location and sketched ideas) should be integrated seamlessly, so that a barrier-free communication is enabled for all participants. The use case is described in Appendix D.1.22.

Contribute Content by Speech Recognition and Gestures

During product development, a number of decisions must be made that allows to have fundamental impacts on the success and cost of a product. Once a decision has been made, it must be captured in the PDM system and communicated to the persons involved. For approvals, this process is reflected by simple forms using multiple choice fields. Voice and gestures recognition could be used as an alternative input method for this kind of simple contribution. The speech recognition can design the release of approval processes more intuitive through simple user dialogs using voice phrases such as "Yes" or "No" or with gesture using "thumbs up" for yes and "thumbs down" for no. The use case is described in Appendix D.1.23.

Contribute Content by filtering captured Objects

Images are meaningful in certain situations as text-based information to describe a subject. Such situations often occur when, for example, the impulse for an idea has to be captured visually (e.g. image of an object and individual drawn digitally sketch by hand) or a faulty product condition has been recognized and needs to be visually commented and communicated. Due to the visually supplemented communication, the reader can extract essential aspects of an image and thus achieve a better spatial understanding of the context specificities. In addition, the captured images can be commented and marked (e.g. elements of the tools palette such as pen, shapes, styles) using simplistic and mobile-suitable image processing tools. For example, a new issue can be created for a produced part using the mobile device to report a breaking point which was caused by heavy loading. In this case, the user would clearly mark the position of the break point with the corresponding image processing tools. However, once other people will be involved in the modification of the part, the scenario must be considered in context of object collaboration, which is described in Section 4.4.6. The use case is described in Appendix D.1.24.

4.4.8 Autonomous Object Actions

The autonomous group includes objects that can act autonomously without human interactions. Decisions are made by objects according to parameters and cognitive behavioral guided structures. For example, measurement data are automatically transmitted to other systems as well as device states at regular time intervals. In this context, no direct dependency to other objects exists, but the object behavior can be adjusted by external influences. Based on various possibilities for autonomous execution of object actions, the following patterns of the cluster have been derived for the Mobile Feature Framework in context of PLM.

Autonomous Transmission of Measurement Data

Mobile devices that have been developed for the purpose of data collection, data analysis and, data transmission are frequently found in locations wherein people reside rarely and they are not permanently present. Such devices operate autonomously and may carry out tasks independently through intelligent system methods of self-optimization, self-configuration, and self-diagnosis. Through cognitive behavioral structures, such cyber-physical systems can be integrated in complex composited seamless networks and can change their own behavior in dependence to other devices. Such devices are used under the aspect of intelligent factory[9] in order to better assist people in increasingly complex tasks. This scenario describes devices with the ability of independent data transmission using IP protocols from the perspective of a central data collection. In addition, the information transmission of intelligent networked objects in the composited network to mobile devices is important with the objective of adaptation and update of behavioral-based structures. An external source sends over the Mobile Feature Framework to the mobile device (e.g. a measurement station with various sensors) the instruction to collect and transmit specific measurement data. The mobile device receives and validates the request from the framework and performs the measurement job under consideration of the predefined parameters. As soon as the measurement results are available, the measurement data are returned to the framework using IP protocols and further processed by mobile web applications or by an external application. The measurement order includes specific parameters when a measurement is be performed permanently or temporarily. The type of the requested data should be transmitted only if a predefined threshold value is reached or exceeded. The measurement data may also include contextual information of the device. The use case is described in Appendix D.1.25.

[9] The objective of the intelligent factory is the creation of adaptable and resource-efficient processes, which are controlled by the product and supported by the use of intelligent networked and modular systems.

Mobile Device Calling Home

In order to capture the vitality of a mobile device over a historical period, a communication link between the mobile device and the central server instance must be established. Such a connection is used to transfer the device state, reports, and diagnostic data in order to plan appropriate preventive measures. The data can be combined with data from other mobile devices to detect and evaluate fault conditions of the entire technical system from an abstract perspective. For this purpose, the data are automatically transferred to the central server instance in regular intervals, so that they can subsequently be analyzed and archived. Thereby, irregularities are detected and appropriate further action is initiated. The use case is described in Appendix D.1.26.

Communication to Adapt the Behavior

Once a cyber-physical system is integrated within complex composite networks, device-specific behavior patterns must harmonize with other devices on the network. Therefore, the interactions of all devices in the entire composited network must be intelligently tuned and coordinated to each other. The synchronization and coordination of contextual device information with the composite network can take place via framework. The exchange of information between devices ensures that no conflicting behaviors occurs. For example, a contradictory behavior exists when the control unit of a system with two devices is showing the following behavior at extreme temperatures:

- Device 1 of the technical system n predicts a critical temperature level
- However, device 1 is not communicating this knowledge, so that all other devices remain ignorant
- Device 2 keeps the actual behavior constantly
- Predicted temperature change occurs
- Device 2 collects the knowledge at a later point itself and must act now abruptly

The result: The contradictory behavior from the perspective of the technical system leads to a rapid change in behavior, which reduces the efficiency (e.g. higher energy consumption) and may affect the lifetime negatively (e.g. intensive wear). The plausible behavior of the entire ecosystem in a composite network is in particular of considerable importance and is composed of the all devices behavior on the network. The communication is performed over standard IP protocols, which in turn requires the communication ability of mobile devices. This use case is described in Appendix D.1.27.

4.4.9 Mobile Feature Combination

As was pointed out in some scenarios, the possibility for the combination of mobile features exists to increase the mobile productivity factor of users. The most presented scenarios not only are serially connected, but can also be connected in parallel as well as intermeshing feature combinations, which consequently lead to a variety of new scenarios. Table 4.2 lists relevant examples of possible feature combinations:

4.5 Development of the Domain Model

The Mobile Feature Framework will be used by three groups of mobile users, stationary users and application developers. The domain model reflects mobile features, which are used for different aspects (e.g. data collection) in workflows and processes that are integrated into mobile applications and provided by mobile devices. The model represents a structure of objects such as features, messages, persons and mobile devices to capture the relationships between objects involved. However, the model does not cover all concepts of the mobile application domain or other related domains, but deals with the core concepts of functional interrelations in the mobile feature domain in context of PLM. The development of the domain model passes through four modeling phases (see Figure 4.5). In the first phase, objects and their relationships are identified and described. The second phase focuses on the persistence of the data. In the third phase, the architecture is split into multiple

Combined scenario 1			
Name:	Shake-based and language-based function call		
	Step 1	Step 2	Target
Description:	Shake the device to activate the voice function	A user-specific commands calls a function	Function call is performed
Pattern:	4.4.1	4.4.7	-
Representation	Shake + Language = Function call		
Mobile features:	Position sensor	Speech recognition	-
Combined scenario 2			
Name:	Location-based voice search		
	Step 1	Step 2	Target
Description:	Determine the mobile user location	Speak a word or phrase	The result set presents objects that are located in the vicinity of the location
Pattern:	4.4.4	4.4.7	-
Representation	Localization + Language = Filtered search results		
Mobile features:	GPS	Speech recognition	-
Combined scenario 3			
Name:	Location-based function call by gesture		
	Step 1	Step 2	Target
Description:	Determine the location to arrange location-specific UI elements	Focus the eyes on a UI element and perform a wink to select it	Location-specific function call is performed by a gesture
Pattern:	4.4.4	4.4.1	-
Representation	Location + Gesture = Function call		
Mobile features:	GPS	Camera	-

Table 4.2: Feature combination on mobile devices

layers and objects are assigned to the corresponding layers. In the fourth phase, the components of the layered architecture are defined and tasks are elaborated. The core of the created framework architecture is not dependent on a specific technology, but dependencies on technologies of the respective mobile platforms exist and will be kept as low as possible. However, the core of the framework architecture can be implemented with various programming languages such as Java, PHP[10] or ASP.NET[11]. Figure 4.6 illustrates the first draft of the domain model and shown the specialized technical features of the subdomains, which have been divided into corresponding demarcated contexts. Each bounded context is isolated and has no knowledge about the internal operations of other contexts. The contexts are connected only by a handful of common identification types such as *FeatureId*, *MobileDeviceId*, and *BusinessId*. DDD provides a set of design elements for the domain model to describe the behavior and states of the domain objects exactly. Each defined bounded context is using Aggregates and Value Objects. The aggregates (in Figure 4.6 illustrated as *«Aggregate Root»*) represent complex hierarchies summarized from Entity Objects and Value Objects that are combined into a single entity and is thus a logically combined entity. In order to avoid direct accesses on individual objects within the aggregate, the access is only possible via *Aggregate Root*. This ensures that the sum of all objects of the aggregate (Entity and Value Objects) is

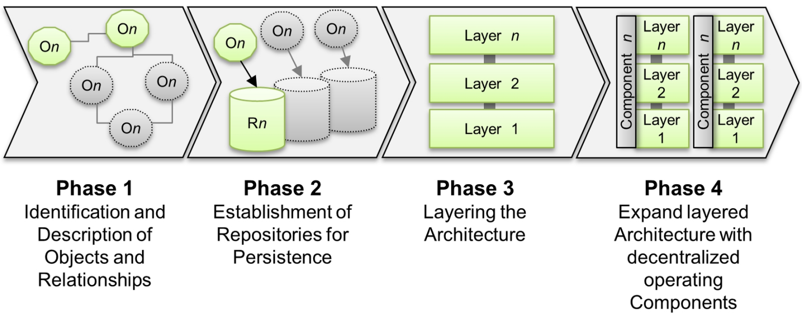


Figure 4.5: Modeling phases

10 PHP: Hypertext Preprocessor (PHP)
11 Active Server Pages .NET (ASP.NET)

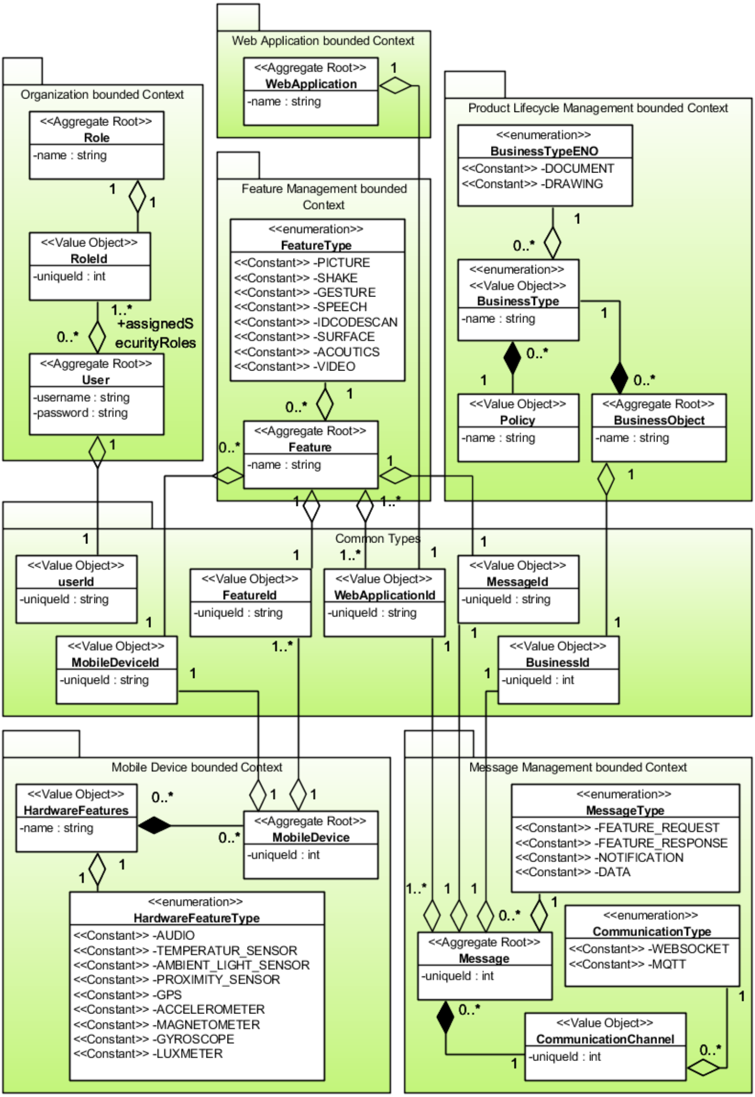


Figure 4.6: Initial domain model

taken into account with all the business invariants during the execution of a transaction. In practice this means that activities of business objects from the PDM system cannot be performed without considering the relevant structures, dependencies, and business rules. Through *Aggregate Root*, the data encapsulation of complex structures are achieved for the respective bounded contexts, which give a better overview of the entire domain model through abstraction, because direct access to internal data structures is not possible unless done only via defined interfaces. The design of the objects feature, mobile device, and communication channel is for the modeling of the Mobile Feature Framework of fundamental importance. The correct and complete behavior modeling of these objects is a fundamental prerequisite for the design of the individual features. Design mistakes that have been generated into the framework modeling would have a negative impact on the feature design and therefore must be recognized and corrected in an early phase. Without the support of specific DDD design patterns, the modeling of feature, mobile device, and communication channel might look like as illustrated in Figure 4.7.

This draft domain model was designed as a single object including the mobile device with features and communication channel. Thereby, the object is accessed via corresponding *getter* and *setter* methods. Nevertheless, problems can still arise as soon as objects are accessed by different contexts (e.g. various mobile PLM applications) and the role-based security concept of PLM is applied. However, generally this simplistic design and modeling does not consider dealing with invoked contexts and various design aspects. DDD offers appropriate design patterns to solve these kinds of problems. Figure 4.7 shows the revised modeling with the classification, mapping, and encapsulation of responsibilities, which are represented by corresponding objects. While mobile devices with hardware-specific features like camera and integrated sensors usually cannot be customized, mobile features can be customized because they consist of software-based logic elements with access to hardware features. Therefore, the hardware features are modeled in the improved design as non-changeable.

In addition, methods that want to change their state, get a new instance of the hardware feature with the respective adjusted state created as well as a hardware feature cannot change its state. For this purpose, the method `useFeature()` is called. This method represents a part of the feature framework backbone in order to model generic use cases of the feature domain. Additionally, by encapsulation, an existing role-based security concept of the PDM system is applied and object states are managed and controlled inside the aggregates. Only immutable objects are exclusively used outside of the aggregate. The second modeling phase considers the persistence of objects. Thereby, DDD concept offers so-called repositories to separate the business logic from the access to the infrastructure. This is necessary to maintain the technological independence. For this purpose, appropriate repository classes are created to abstract and encapsulate the persistence technology (e.g. functionalities like search, load, save and remove of objects). Repositories are the only components of the technical model with the ability to access objects of the infrastructure layer. This layer has its own design patterns. As a suitable design rule in the DDD modeling, the creation of a repository per aggregate has become established. Based on the first modeling phase, repositories are now

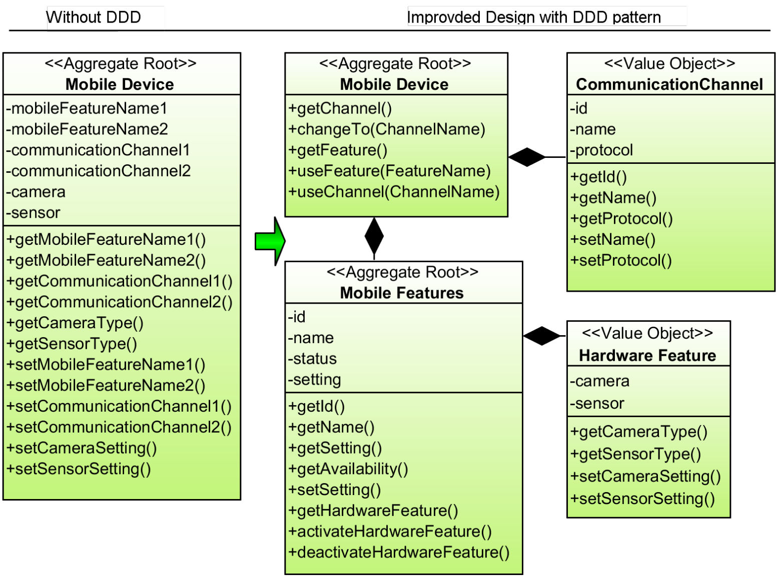


Figure 4.7: Modeling phase 1: Initial design vs. improved design

created and assigned to appropriate aggregated objects. Furthermore, the objects are assigned to the corresponding layers. The layer-based architecture of the domain model is described in the Section 4.6. Firstly, it is sufficient to understand that a client represents a generic resource to access appropriate services from the feature framework. A framework service receives and translates requests from the client for the domain-specific application case. The framework service loads the associated aggregate from the repository and calls the appropriate function of the aggregate. The totality of all the components involved from the different layers represents the fundamental building blocks in order to perform a transaction. Each new client requests accordingly results in a new transaction, which affects other objects depending on the request. The management and control of transactions are not the responsibility of the domain model, because the transaction management includes specific technology-dependent aspects. Therefore, the logical components of transaction-specific concepts are decoupled from the domain layer and managed separately in appropriate repositories. The result of the second modeling phase is illustrated in Figure 4.8. In Section 4.6, the current domain model will be further developed in terms of a layer-based architecture in order to realize the potential of modular components for the framework.

4.6 Layering the Mobile Feature Framework Architecture

The first two modeling phases have been already carried out in the previous Section 4.5. Objects were identified, subsequently combined to aggregates, and appropriate repositories for the persistence were created. Due to the complexity of the domain and the resulting benefits of modular components, the architecture of the Mobile Feature Framework will now be divided into several layers. The respective layer assumes various duties and responsibilities as well as requires an appropriate perspective on data objects and interactions. Thereby, each defined context (see Figure 4.3) reflects a separate segment, which contains the components of the layered architecture. The user interface layer is responsible for the presentation of information as well as for the interpretation of user inputs. The application layer coordinates the tasks within the application and stores the current status of tasks that were performed by mobile users. The domain layer contains the business domain of the bounded context and the business objects. However, this layer is not responsible for the persistence of business objects but therefore delegates the storing to the infrastructure layer. The infrastructure layer implements the persistence for business objects and acts as a repository for all the other layers by providing components for communication, security, logging and exception handling as well as transaction management.

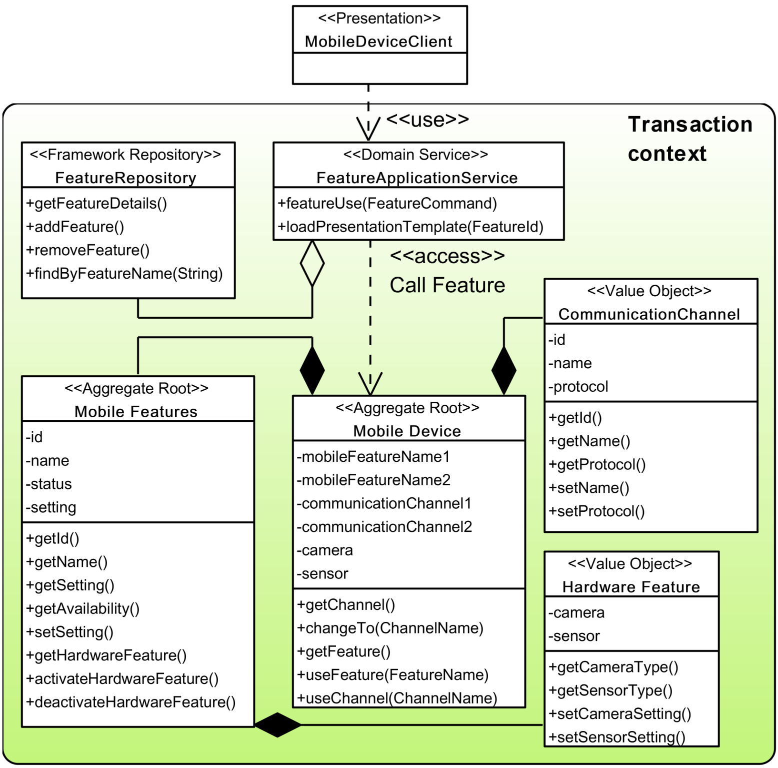


Figure 4.8: Modeling Phase 2: Design with repositories

- **The presentation layer** does not contain any logic components and business objects. Only view components isolated from each other are provided and only formed to navigable workflow views through the application layer. In addition, this layer does not validate inputs of mobile users, but focuses on the presentation of information on mobile devices and input opportunities for mobile users. Input opportunities could be based on textual, visual, acoustical, location-specific, or device-specific inputs.
- **The application layer** defines tasks for corresponding scenarios to cope with identified problems with the help of deeper layers. For this purpose, all activities of the presentation layer are coordinated by views, which are connected to a sequential series of operations and thereby a full-fledged workflow is formed. The presentation logic is reflected in the application layer, which includes the navigation between the user interface (UI) views and simple non-business-related validations. For example, the validation includes the verification of dates and special characters used in input forms. In doing so, the progress of performed tasks of mobile users is captured, but the state of business objects and business rules are not stored in this layer. Business object related tasks are delegated to the underlying domain layer, which also determines the state of business objects.
- **The domain layer** comprises all information about the business domain, which reflects corresponding use cases through a concept. The use cases are represented by generic patterns through business logic and business rules. The business logic consists of business objects with the associated conditions. This layer also provides the validation of business-relevant processes and states by a set of rules. The persistence tasks for business objects are thereby delegated to the underlying infrastructure layer.
- **The infrastructure layer** supports the overlying layers by basic functionalities like the abstraction of persistence for business objects (such as database or file-based repository). For this purpose, the access methods of the business logic are separated from the technical infrastructure to achieve a high degree of isolation. In addition, the layer provides appropriate functionalities for the transmission of messages between the framework and mobile devices as well

as repositories for storing view components. Following the now completed modeling, the results are shown in Figure 4.9.

The result of the layered architecture has the consequence that all layers are able to operate independently. For example, the exchange of user interface components which have been implemented with Java Server Pages (JSP) and are now to be successively replaced or modernized with PHP, does not affect the business logic. Similarly, the change of the used technology to persist data has no effect on other layers. For example, the database technology is replaced from Oracle database to DB2. The respective impact of changes is related to the corresponding layer. Section 4.7 describes the last step of the modeling phase, which contains the division of decentralized operating components of the layered architecture.

4.7 Components in the Mobile Feature Framework

Through various location-based and technologically related aspects of the particular contexts, the creation of decentralized operating components is essential. Therefore, it is necessary to realize the corresponding pieces of the layered architecture in different contexts. For example, it is essential to provide appropriate components on mobile devices to access mobile features, which enable the possibility to interact with other components within the framework. However, the core of the business logic is provided by central components, while the access is performed using standard access methods as well as corresponding transport and security protocols. The basic components of PDM systems cannot cover these functionalities for the Mobile Feature Framework. Therefore, the framework components for integration of mobile features must be provided outside of the PDM system. Firstly, the determination of required components has to be conducted from an abstract perspective. As a base for this, the already identified contexts are considered. Mobile features that are provided by mobile devices have mostly their own proprietary mobile platforms. In addition, mobile devices need appropriate components for the feature access due to the

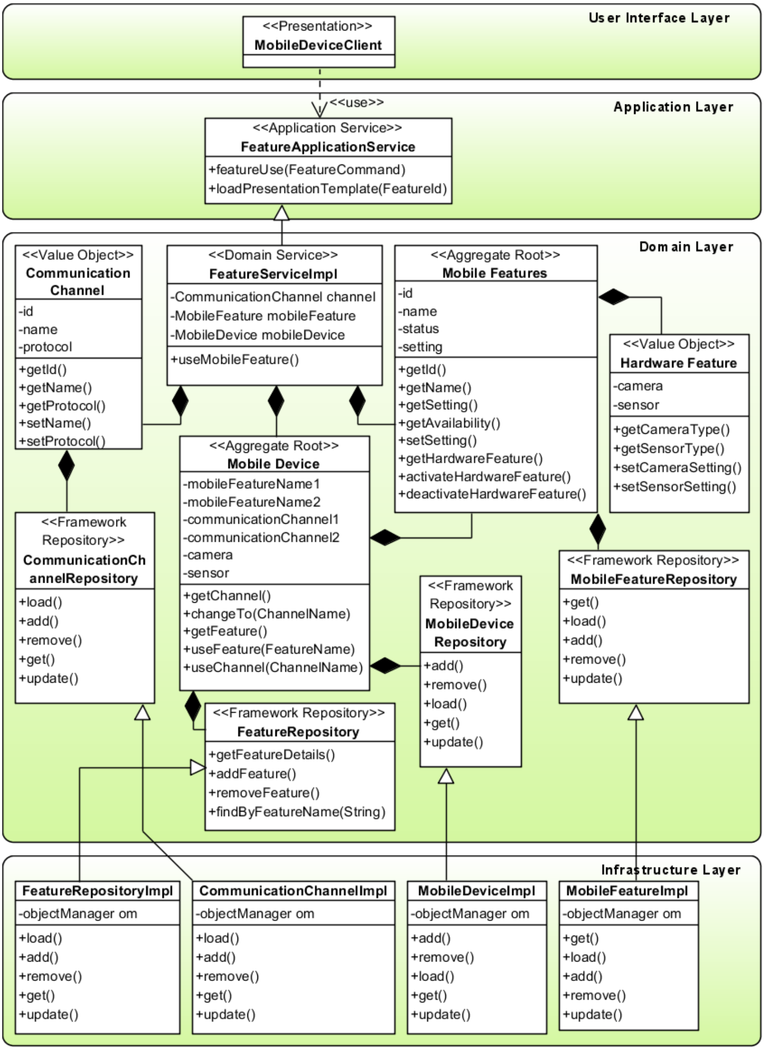


Figure 4.9: Modeling phase 3: Design with layered architecture

fact of being "mobile". Therefore, a separate abstractly defined and generic component is intended. This component is described as *Mobile Device Client* (MDC). The MDC component is explained more precisely in Section 4.7.3. From a technological point of view, web applications can be realized as client-side and server-side components. However, the expectation must meet the requirements to manage and control all mobile PLM applications through a central instance. From the design perspective, these conditions cause for client-based components of the web application context corresponding disadvantages. The drawbacks of the architecture would create an additional communication overhead between clients and servers, difficult reusability of components, increased maintenance costs as well as a complex and partially imperfect interoperability. Therefore, a separate server-side component is intended for web applications and is called *Mobile Device Application* (MDA). The MDA is described in detail in Section 4.7.4. The contexts PLM and organization represent the underlying PLM concepts and organizational structures. Both contexts already exist in the stationary current IT landscape. However, PLM concepts and organizational structures are more or less pronounced and are not part of the business logic. Accordingly, both contexts are reflected by the component *Mobile Device Integration* (MDI). The MDI is described in Section 4.7.2. The context of the Mobile Feature Framework reflects the business logic and is used as core element for communication between other components. The business logic comprises all service elements, which are accessed by client components of the MDA to apply a corresponding use case. The use case involves all required objects to perform the transaction. For this purpose, a separate component is provided for all centrally controlled and manageable tasks of this context as well as is considered as core of the architecture. This component is described as Mobile Device Framework (MDF) in Section 4.7.1. In summary, four generic components have been identified and defined for the framework architecture including the *Mobile Device Client* (MDC), *Mobile Device Framework* (MDF), *Mobile Device Application* (MDA) as well as the PDM integration *Mobile Device Integration* (MDI). The *Mobile Device Manager* (MDM) and *Mobile Device Samples* (MDS) are optional components, because they are not essential for the framework operation and the framework can be used as a unit without these components. The MDM and MDS are primarily designed for administrative and demonstrative purposes, which are discussed in Section 4.7.5 and 4.7.6. As a result of the specified components, the architecture without considering external used components of the framework such as Message Queuing is shown in Figure 4.10. Section 4.7.1 to 4.7.6 discuss the objectives and cross-divisional tasks of the identified components in more detail. This will be followed by the integration of the distributed components in the domain model. For this purpose, a separate layered architecture for the presentation of UI elements, provision of application services, domain logic, and data persistence is provided for each component. The objects in the components of the layered architecture respectively interact with objects in the components of other architectural layers. The domain model, which considers the distributed components with related objects, life cycles and interactions with other components is described in Section 4.8.1 to 4.8.4.

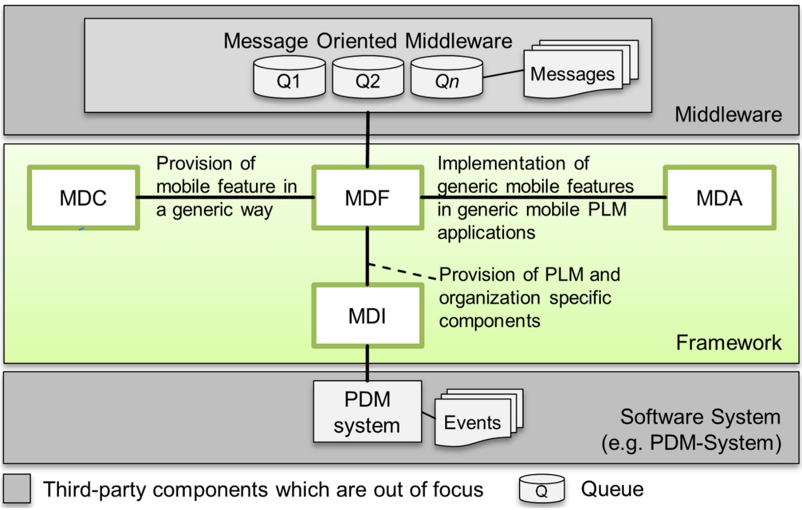


Figure 4.10: Components in the architecture model

4.7.1 Mobile Device Framework (MDF)

This component represents the core of the framework. The server-side operating framework acts as link between the platform-specific Mobile Device Client (MDC) and the server-side framework components such as the integration module (MDI) and Mobile Device Application (MDA) as well as Mobile Device Manager (MDM). The MDM and MDA are accessing the interfaces provided by the framework to perform interaction with mobile features on the mobile device, query business objects, and configure framework-specific components. The interface access is performed via web service in order to realize mobile applications across platforms in any programming language. To ensure that the communication of the MDC application is running on the mobile device, the framework offers various communication channels and network protocols such as WebSockets and MQTT. The communication link depends on the boundary conditions such as the latency and power usage of the network protocols, the total amount of data and number of data packets, which have different performance characteristics. The TCP-based WebSockets protocol establishes a bidirectional link between MDC and the MDF. While the MDC only includes a WebSocket client module, the MDF contains the modules for a WebSocket server and WebSocket client to ensure the mutual exchange of information. During the initialization phase of the MDC application, the built-in WebSocket client establishes a general connection to the WebSocket server of the MDF to permanently use this open connection for the exchange of information without the need to set up each time a new connection. The exchange of information takes place in both directions. The Mobile Feature Framework supports the approach of the publish-subscribe communication on which messages are sent to a specified topic. Additionally, an unlimited amount of recipients can consume these messages. The WebSocket client of the MDC subscribes to a fixed *Topic* in which the MDC sends and receives appropriate messages. The WebSocket server instance of the MDF has registered the *Topic* for communication and is actively listening for incoming messages as well as sends to the topic appropriate messages, which in turn are consumed by the MDC. The *Message Queue Telemetry Transport* (MQTT) is provided by the MDF as an alternative protocol for message exchange with restricted networks to allow the transmission of telemetry data and feature data. Such restricted networks are usually limited in the total amount of data to be transferred and have mostly high latencies, which make it difficult to reliably operate applications that were originally designed for wired networks. MQTT also supports the publish-subscribe model for the message exchange as in the case of WebSockets. JSON is used for the data exchange between the MDF and MDC. This structure is a text-based data format that can be read by almost any programming language to ensure the independence of the communication. In addition, JSON reduces the overhead through a readable and compact syntax compared to the XML format. The low overhead is particularly interesting under the aspect of mobile networks, because a small amount of data simultaneously means a better performance of the mobile application. Thus, data are more quickly available at the destination and excessive waiting times are reduced accordingly to a minimum. However, JSON does not support all data types and therefore requires alternative solution for the conversion and use of other data types. The framework provides a JAX-WS-based web service for the interaction between the MDA/MDM and the framework component MDF, which must be implemented by the respective components. The usage of web services requires the authentication by the MDA. In case that the authentication is successful, a login token is allocated. The login token is an identifier that consists of a 16-bytes hexadecimal number and is divided into five groups. The login token remains valid until the session of the MDA or MDM is terminated. The authentication is performed by an external directory service using the username and password of the user account. To allow users to start mobile applications (MDAs) using the MDC application, they must be registered against the MDF. The registration process is coupling the MDA to a communication channel over which the MDA communicates with the MDC. The communication channel must be previously defined by the MDF, which can either be a WebSocket server instance or a MQTT server instance. In addition, parameters are assigned during the registration process that defines whether specific access point to the MDA, and whether the MDA should be available at all times on the MDC. The MDF assumes the responsibilities of the user authentication. Therefore, the MDC transmits during the initialization phase the username and password to the MDF. Subsequently, the framework verifies the defined method of authentication. The authentication procedure can take place over a local or remote directory service. The external directory service uses the *Lightweight Directory Access Protocol* (LDAP), which performs the query against an LDAP server. LDAP has been implemented as a classic client/server model so that a client-side request is replied with a corresponding server-side response.

4.7.2 Mobile Device Integration (MDI)

In this work, the integration component is understood as a module that manages the communications to an external data backbone. The focus of the integration module is on PDM systems, even though further integration modules of other domains such as ERP and CMMS may also be possible. The PDM integration module is responsible for accepting generic requests, which have been created by mobile applications and forwarded by the framework to transform the generic inquiry into the PDM system typical language. The system-specific response of the PDM system returns the corresponding results back to the framework. The results are subsequently returned to the mobile application in a harmonized and generalized form. The integration module uses the JAX-RS web services implementation for the communication to the PDM system ENOVIA to the exchange of data objects and JSON as data format. The challenge of this component is to understand the vendor-specific interfaces. Depending on the PDM system, business objects and file objects must be called by the MDI in different ways so that mobile applications must not implement any native or proprietary interfaces of the manufacturer. The complexity of the request transformation between mobile applications and the PDM system grows steadily with increasing cross-linked objects, so that simplified workflows have some benefits under the aspect of mobility.

In addition, the majority of mobile users would like to use mobile applications on mobile devices that are intuitive usable as well as provide less detailed information and extensive functions.

4.7.3 Mobile Device Client (MDC)

The platform-specific client has the primary task to ensure the interaction between the mobile device and its associated device features. The client can be triggered using three defined options. The first option, which recognizes the incoming events and executes configurable actions, represents a platform-specific listener within the client. Thereby, it is possible to use mobile features outside of an active workflow context in order to initiate appropriate interactions. The second option represents the interactions between the mobile user and a workflow as well as the resulting actions based on the user interface. The mobile features are integrated into a workflow by a server-side running mobile application and feature interactions are controlled via feature messages. The results of features are returned back to the mobile applications using a defined reverse channel in order to react in the actual performed workflow. The third option constitutes server-side triggered events, which are transformed into feature messages and transmitted to the client over established and active communication channels. The incoming messages are interpreted by the MDC client and subsequently, corresponding event-depending actions are performed on the mobile device. Due to the variety of mobile platforms, the access to native mobile features can be achieved only through platform-specific clients, even though the demands of a cross-platform client are high. HTML5 offers the opportunity to access a small number of mobile features, but the possibilities of interactions are very limited and the varieties of features are restricted. For these reasons, a native implementation of clients per mobile platform is necessary in order to ensure the diversity of the feature set and to provide enough space for innovate feature interactions. The client should only provide the minimum functionality, which is necessary to enable the feature interaction, but not to depict workflow components and processes. The representation of workflows and the composition of mobile features should continue to be designed and executed on server-side components. The interaction with platform-specific features is one of the core competencies of mobile clients. The feature messages describe the feature type, interaction type as well as procedure with generated data which result from interactions. Once the MDC application is started on the mobile device, all defined features are loaded by the plugin manager and associated configurations are verified for each feature. The life cycle of the plugin is defined by the plugin manager. The plugin is responsible for the interaction with the mobile feature. The plugin contains basic service functions like starting, pausing and, stopping of features as well as the creation of a response including the feature states and results, which are returned to the server-side framework component once the user has finished the interaction with the feature.

4.7.4 Mobile Device Application (MDA)

The server-side executed mobile application represents the software component, which is used by the feature framework and must be implemented by application developers. The application is responsible for the entire workflow of the mobile user and combines sub-components to be used within the workflow. The mobile feature access is completely transparent for the applications. The communication to mobile features takes place over a web service interface that is provided by the framework. During the execution of a workflow, the MDA application requests a corresponding feature in the feature framework using a feature message, which is executed with specified parameters on the mobile client. Once the feature is called by the MDC, the component takes over the interaction with the user. After completing the feature interaction on the mobile device, the continuation of the workflow is initiated by the MDC so that the MDA can continue the workflow. The MDA receives the resulting feature data from the MDC for evaluation and integration purposes.

4.7.5 Mobile Device Manager (MDM)

This component is server-side executed and assists users to perform administrative tasks. These include the definition and administration of communication channels, the registration of mobile applications as well as the assignment of communication channels to mobile applications. This application accesses the webservice-based interface of the Mobile Feature Framework and visualizes the configuration options through a user interface specifically adapted for mobile devices.

4.7.6 Mobile Device Feature Samples (MDS)

This component includes reference implementations that demonstrate the capabilities and functionalities of the feature frameworks for testing purposes. The MDS is particularly suitable as a starting point for application developers who have to develop MDA applications using the feature framework. This component contains comprehensive implemented code examples, which demonstrate, for example the usage of the camera, audio, map services, and barcode scans that have been requested through the user interface, the communication to the framework as well as the data analysis for the workflow processing.

4.8 Building Blocks and Services in the layered Architecture

This part covers the business logic modeling for the identified components of the framework architecture based on Section 4.7. For this purpose, the created subject-specific models of the bounded contexts are reused from the previous sections as a basis for component-based modeling of the subject-specific models. In addition, the subject-specific models of the framework components are elaborated in detail and complemented by additional constituents of DDD.

4.8.1 Modeling the Mobile Device Context

The architecture of the mobile device context has all four architecture layers. Figure 4.11 shows a simplified representation of modeled context. The `ViewController` of the UI layer is responsible for the composition of native view elements. Native view elements are understood as specific UI components of the respective mobile platform. The user interacts through user input with the `ViewController`, which subsequently calls the appropriate service in the application layer (see Table D.41). In addition, the `ViewController` carries the stored view information over the associated repository from the infrastructure layer. The result of the composition made up of business data and views are presented to the user by the ViewController. The *Mobile Device Client* represents the core element of the domain layer and is composed of the ClientCommunicationChannel for incoming communications and the aggregate MobileFeaturePlugin. The ClientCommunication Channel retrieves the information from the repository for setting up the communication channels. Thereby, the interface ClientCommunica tionChannelRepository ensures that the specific implementations are decoupled from the domain layer and realized in the infrastructure layer. The infrastructure layer establishes the connection between the logical and independent domain layer with the specific infrastructure layer. Accordingly, the low-level tasks of the transport channel are delegated to the infrastructure layer, which is represented by the Trans portService. The collection of objects of the aggregate Mobile FeaturePlugin contains the Value Object HardwareFeature, which does not have its own identity. The object HardwareFeature is solely defined by the hardware characteristics of the mobile device. Since the hardware features of the mobile device do not change, the object is simultaneously modeled as immutable object. Once the Hard wareFeature object has been created, it can no longer be changed or modified. In addition, with the MobileFeaturePlugin, information about plugins is stored in the repository and loaded over the interface Mo bileFeaturePluginRepository using the associated object in the infrastructure layer. The MobileFeatureClientService is used for the generalization of feature services. For this purpose, a Feature RequestEvent is triggered by the MDF as domain event, which in turn starts the initialization of the feature process. As a result of the feature processing, a FeatureResponseEvent is created as domain event, which in turn triggers the MDF. This event includes a set of various elementary and complex data types. The domain event Data Event is also instantiated by a feature request. However, a single response message is not sent, but rather a continuous data stream over the communication channel. The data sets are provided by the corresponding feature. The triggered events in the form of sent and received feature messages are stored persistently in the EventStoreRepository using the interface EventStore. The JMSMessaging reads pending events from the EventStoreRepository and tries to transmit the message using the underlying messaging infrastructure, as long as the delivery

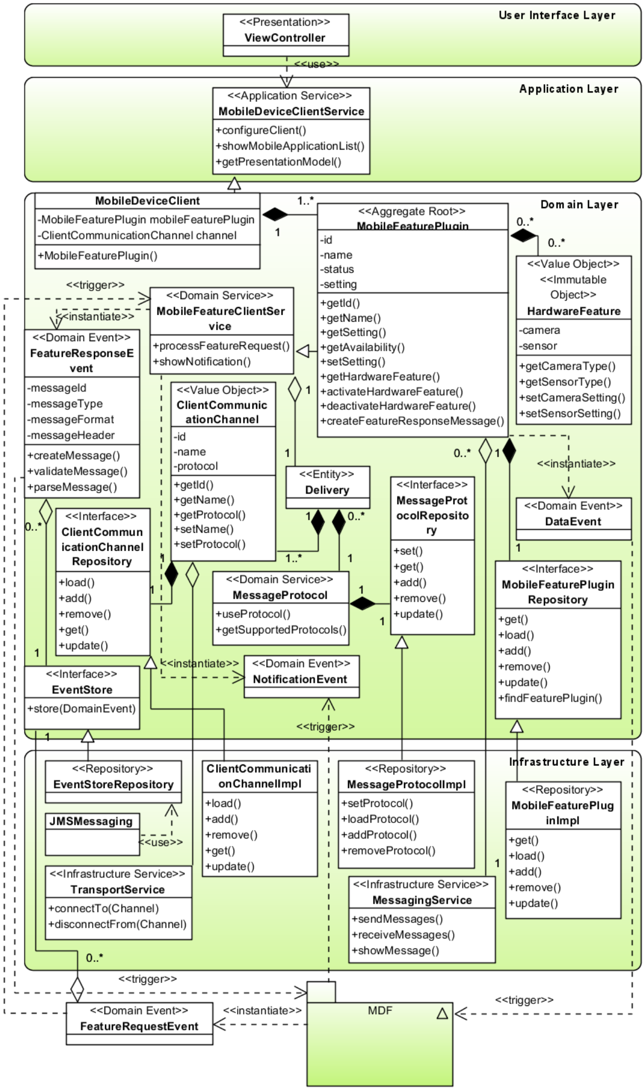


Figure 4.11: Domain model for MDC

of the message to the target system (MDF) is successfully completed. The message delivery is performed through a separate transaction. The incoming and outgoing feature events reflect various types of messages, which depend on the state of life and vitality of the individual features. A faultless feature execution would thus trigger another event type as an erroneous feature execution. However, the inherited event as a generic and derived component always remains constant. The `MessageProtocol` retrieves relevant information from a specifically created repository and depends on the respective entity `Delivery` as well as the communication channel `ClientCommunicationChannel`, which has no conceptual identity. The `MessageProtocol` has been modeled as a domain service, because message protocols can be used depending on the mobile context in order to reduce the message overhead at low bandwidths through appropriate messaging protocols. In summary, a series of events, services, and repositories have been identified during the component-based modeling for the bounded context of mobile devices, which are listed in Appendix D.3, Tables D.41, D.45, and D.49.

4.8.2 Modeling the Mobile Application Context

The context of the mobile application has four architecture layers (see Figure 4.12). Because the major parts of the business logic are modeled in the MDC and MDF, the domain layer of the MDA is clearly characterized only by few domain events to and from the MDF. The `ViewController` of the UI layer interacts with the user through user inputs as well as is responsible for the composition and provision of business object information and web-based view elements. Web-view elements are understood as UI components, which are used independently of mobile platforms. The `ViewController` interacts with the `ApplicationService` of the application layer to call the domain logic and to retrieve corresponding business object data. In this regard, the respective views are loaded from the `ViewRepositoryImpl` of the infrastructure layer using the associated repository interface `ViewRepository` of the domain layer. Thereby, the interface `ViewRepository` guarantees the decoupling specific implementation of the infrastructure layer from the domain layer. The central object of the domain layer is the `WorkflowObject` that uses

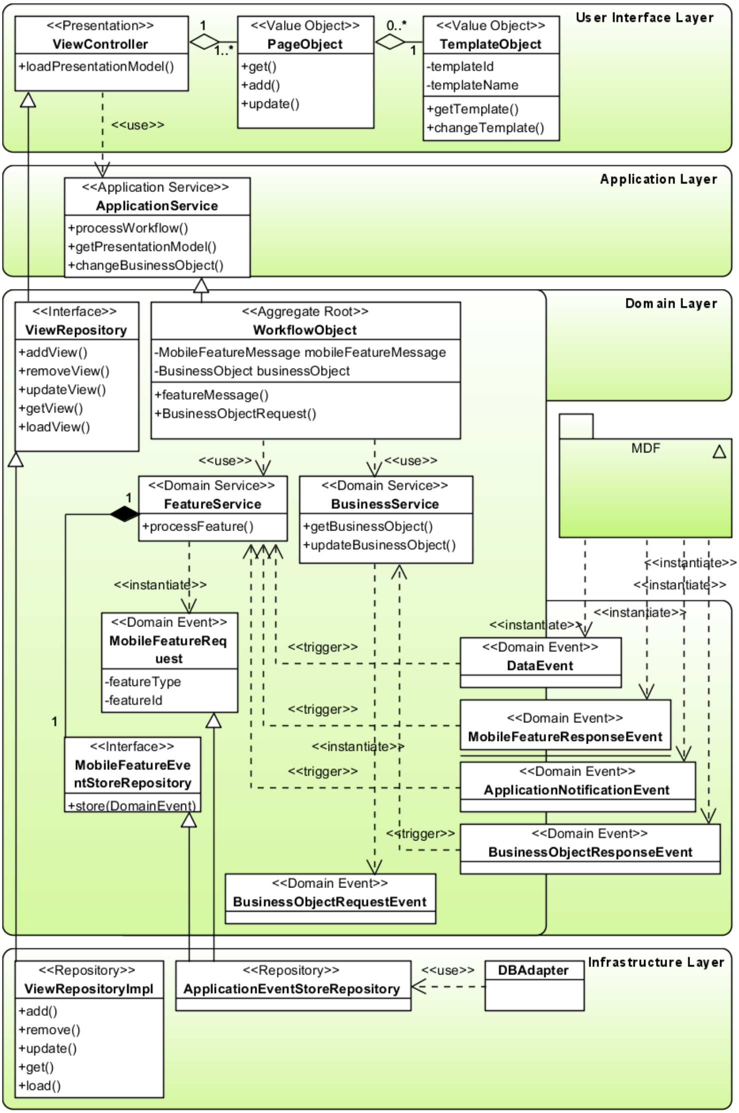


Figure 4.12: Domain model for MDA

the services FeatureService and BusinessService. The FeatureService is used for creating, validating, managing and, storing feature requests and feature responses. The FeatureService has the task to initiate the process of feature interaction by creating a MobileFeatureRequest event, which is forwarded to the MDF. The MDF manages the event processing to the platform-specific MDC. After completion of the feature interaction, the result of the MDC is transmitted over the MDF to the MDA as an event of type MobileFeatureResponse. The MobileFeatureResponse includes from the plugin the results of the feature interaction which are subsequently evaluated and processed by the MDA. For this purpose, the events are stored in the ApplicationEventStoreRepository. The DBAdapter takes the events from the ApplicationEventStoreRepository in order to save these persistent in a suitable database. In addition to MobileFeatureResponse, there are events such as DataEvent, ApplicationNotificationEvent and BusinessObjectRe sponseEvent. The DataEvent is triggered by the MDF through a previously initiated MobileFeatureRequest, when data are transmitted continuously from the MDC to a server-side instance. Once the requested information is available as a continuous data stream, the DataEvent is triggered from the MDF to the MDA. The ApplicationNotificationEvent is delivered from the MDF to the MDA, but this event type is triggered and processed in the PDM system over the integration MDI or by the MDC. In this case, the framework component MDF is responsible for the type of notification delivery. The incoming ApplicationNotificationEvent triggers the FeatureService, which interprets the event and initiates further action. The BusinessService creates, validates, and stores BusinessObjectRequestEvents, which are forwarded to the integration component MDI. The MDI determines the requested business object and provides to the MDA a copy of the business object data. It is noted that the MDA works only with a copy of the business object, but never on the original business object. The business object data are modeled as BusinessObjectResponseEvent that triggers the BusinessService. The BusinessService is now able to integrate them into workflows for further processing of the business object data. All services, repositories and, events used by MDA are listed in Appendix D.3, Tables D.44, D.46, and D.50.

4.8.3 Modeling the Context of the Framework Core

The context of the framework core has only three architectural layers. The UI layer presented in the MDC and MDA architecture is not needed by the framework core architecture, because the main focus is on the business logic of the modeling. All services, repositories, and events used by the MDF are listed in Appendix D.3, Tables D.42, D.48, and D.52. Figure 4.13 shows a simplified representation of the domain model. The application layer has the services `MobileDeviceService` and `MobileApplicationService`. The `MobileDeviceService` accesses the `MobileFeatureService` of the domain layer, which uses the aggregate `MobileDevice` consisting of combined entities. In the modeling of the framework it is assumed that mobile features are always a part of the mobile device and thus all the communications and controls are performed over the mobile device. Therefore, the `MobileFeature` is modeled as part of the `MobileDevice`. The abstraction and definition of mobile features are represented by the aggregate `MobileFeature`, which has a relationship to the interface `MobileFeatureRepository` so that the specific and proprietary mobile features can be modeled decoupled from the business logic. The interface `MobileFeatureRepository` generalizes the specific implementation of the respective mobile platform, which is represented by the `MobileFeatureImpl` in the infrastructure layer. The `MobileFeatureService` instantiates the `MobileFeatureRequestEvent` as event type of the domain logic in order to describe the technical occurrences and to initiate actions between the MDF and the mobile client (MDC). Thereby, the `MobileFeatureRequestEvent` is triggered by the MDA and sent to the MDF. The MDF processes this event before it is forwarded to the MDC on the mobile device. After completion of the feature interaction, the MDC triggers a `MobileFeatureResponseEvent` as a result, which is transmitted and subsequently processed by the MDF. The `Delivery` ensures that the appropriate communication channel and the suitable message protocol are used for message transmission. The message delivery is not performed by the `Delivery` but is carried out by the `MessagingService` of the infrastructure layer. The `MessagingService` is modeled as a low-level service and implemented in the

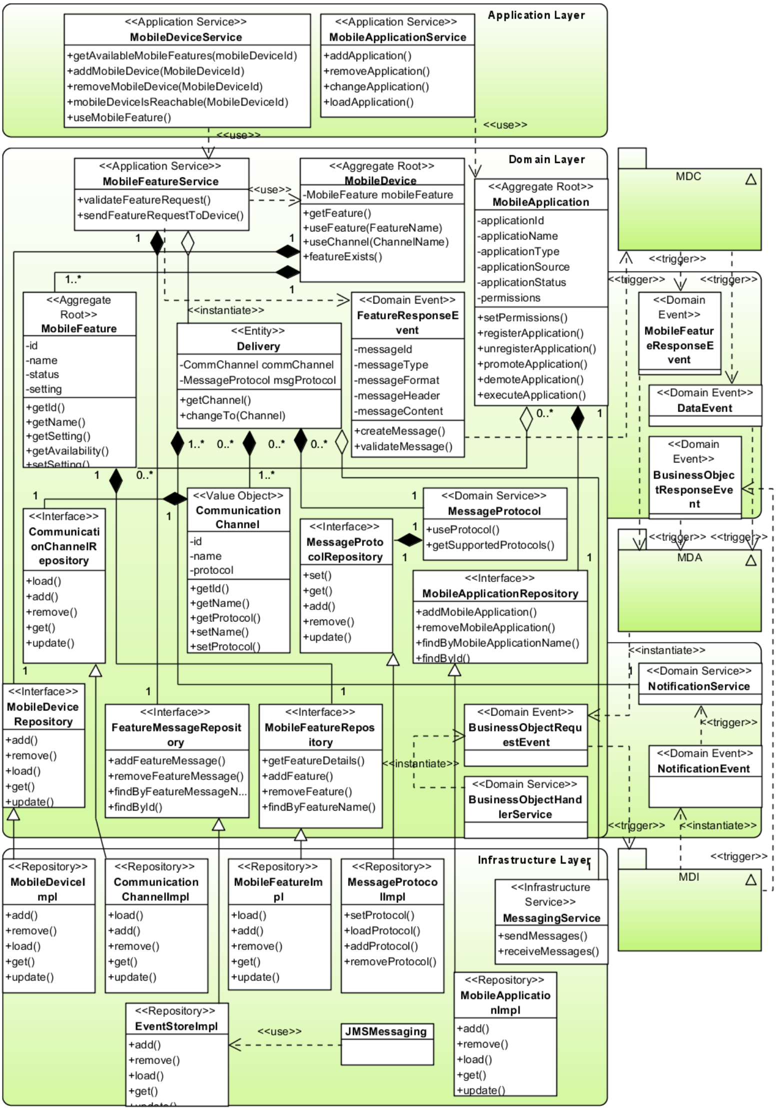


Figure 4.13: Domain model for MDF

specific messaging technology. In addition, the MDF is triggered by the MDC through event `DataEvent` to coordinate continuous data streams from the MDC to the MDA. The MDF registers the incoming `DataEvent` from the MDC and invokes the *Delivery* of the framework in order to select an appropriate delivery method so that the data stream can be received by the MDA. The delivery of the data stream uses a specially defined and secured communication channel of an implemented message queuing system[12] on the infrastructure layer for transport. The triggered event `BusinessObjectRequestEvent` of the MDA is processed by the framework and delegated to the integration MDI. Once the integration module has retrieved the data from the PDM system, a corresponding `BusinessObjectResponseEvent` is triggered by the MDI to the MDF, which is subsequently forwarded to the MDA. Events can be registered by the MDI, whenever they occur in the PDM system. For this purpose, the MDI triggers the event `NotificationEvent` to the MDF. The MDF registers the event and uses the `Delivery` again in order to perform the message delivery over the corresponding communication channel. The `MobileApplicationService` of the application layer uses the aggregate `MobileApplication` of the domain layer to add and remove applications. The `MobileApplication` has a relationship with the `Delivery`, because the delivery of events by the MDF varies from the respective mobile applications.

In addition, the `MobileApplication` has access to the repository using the associated interface `MobileApplicationRepository` in order to store, update, and load data persistently in the specific repository implementation `MobileApplicationRepositoryImpl`.

4.8.4 Modeling the PDM Integration Context

The PDM integration context has three architectural layers without the UI layer, because no view elements are required in this layer. Figure 4.14 shows a simplified representation of the domain model. All services,

[12] Message Queueing systems are part of the Message Oriented Middleware (MOM) software solutions for sending and receiving messages between cross-platform distributed system architectures.

repositories and, events used by the MDI are listed in Appendix D.3, Tables D.43, D.47, and D.51. The MDA instantiates the event `BusinessObjectRequestEvent`. This event is forwarded over the MDF to the MDI and registered by the integration. The domain service `BusinessObjectService` of the MDI processes the event by querying requested data of business objects from the PDM system. Therefore, a `BusinessObjectResponseEvent` is instantiated and transmitted to the MDF. The `BusinessObjectService` has a relationship to the aggregate `BusinessObject`, which has no knowledge about specific business object implementations of the respective PDM systems and thus all entities around the `BusinessObject` are modeled generically. A mapper ensures that business objects of the legacy system are transformed into a generic `BusinessObject` before the information is returned to the MDF. In this case, all activities are always performed on a copy of the original data. In order to avoid that the subject-specific model of the framework is affected by PDM-specific models, a separation of the two models must be done by an anti-corruption layer. The anti-corruption layer connects the contexts mobile device and application with PLM and organization. For this purpose, the mobile feature concept and the associated elements are translated to the PDM model to provide generically legacy information of the target system for other layers. The objects of the target system are transformed into the corresponding domain objects through a mapper. Therefore, the anti-corruption layer of the MDI presents a set of services used by the MDF at the application layer. The communication with the target system is completely transparent for all layers outside the anticorruption layer. On the one hand the anti-corruption layer is not an additional layer, but on the other hand a DDD pattern consists of a set of messaging services in order to keep two bounded contexts in sync. The result is an independent domain model and data model, which have a lot of integration elements and act as a link between these models. The integration elements are represented by various building blocks of DDD for example, repositories and services. The translators communicate with the respective components in both directions (direction to MDF and target system such as PDM system) to support a bi-directional anti-corruption layer. Thus, ORM-related implementations are consistently kept outside the domain model. Messages, which are triggered in the PDM system by respective user actions, are registered as a `LegacyNotifi`

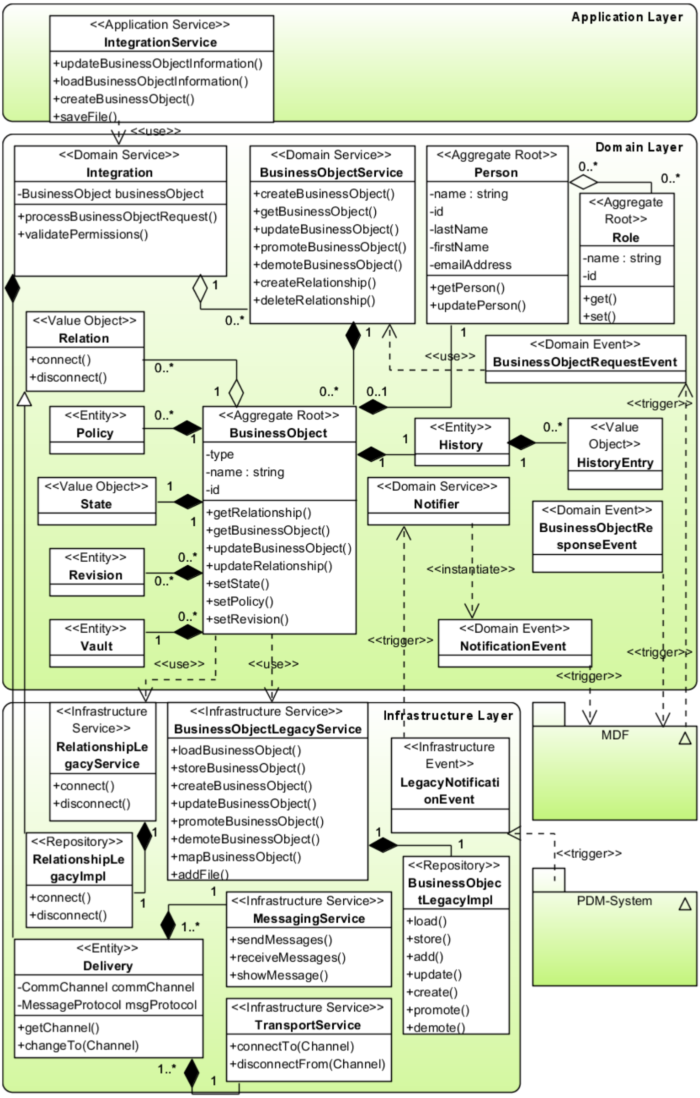


Figure 4.14: Domain model for MDI

`cationEvent` of the specific PDM implementation and communicated to the `Notifier`. The `Notifier` translates and transforms the specific message into a generic `NotificationEvent`, which is transmitted to the MDF. Subsequently, the MDF performs the delivery of the notice. This ensures that specific PDM implementations are decoupled from the business logic. In essence, the system integration model consists of the following three components (see Figure 4.15): Perception and processing of occurring events, query of objects (information and files), and direct integration of components into the target system.

Perception and processing of occurring events

Events are triggered outside of the framework core. Therefore, events are accepted from the appropriate integration module and processed according to predefined guidelines. The perception of events can take place from two directions. On the one hand, the event listeners of the integration module can use the API of the target system in order to recognize occurring events. On the other hand, the triggered events can be perceived by an integral component in the target system while a trigger of the target system invokes the integration module of the Mobile Feature Framework. This method is dependent on the respective target system and its supplied functionalities.

Query of objects (information and files)

In order to allow MDA applications to retrieve certain data objects over the integration module from the target system (e.g. PDM system), the MDI component of the framework must implement a data collector. The tasks of the data collector are accept requests from MDAs to transform requests into the language of the target system, deliver requests to the target system, and return datasets back to the MDA via MDF. The realization of this procedure can be done mainly through vendor-specific implementation caused by proprietary target systems.

Direct integration of components into the target system

Appropriate proprietary components of the target system, which are connected to the framework over the MDI can be implemented if the manufacturer of the target system permits integral components. Therefore, elements are integrated into content-based structure components (e.g. menus, navigations, and commands) as well as into work processes (e.g. workflows, lifecycles). For this purpose, the user's context must be always considered, i.e. whether an interaction is carried out with mobile or stationary users. Thereby, specific functions are in dependence to the user context. For example, a stationary working user can send messages to a mobile user and can determine the actual location of the mobile user via framework. Thereby, user-specific information can be used in order to prepare the user's data for the appropriate context and to obtain user's feedback about specific topics. Thus, mobile users are integrated into the target system as a significant team user group and accordingly noticed more active by stationary users.

4.9 Overlapping Submodels in the Framework Architecture

Different points of view are taken and captured by various sub-models to consider the framework not only from the modeling perspective of business logic. The main emphasis here is on how further requirements can be taken into account in the modeling. These further requirements were not covered by the already designed business logic. Therefore, a consideration of messaging, security as well as data privacy and user directory will take place from the particular perspective.

4.9.1 Messaging Model

Messaging model has the responsibility to consider all design aspects with respect to the message and message transport. This includes the message construction in the appropriate format, the message transformation for the transport including the validation and consistency check as well as the message delivery over defined communication channels. Messages can arise in various aspects at different locations both within and outside the framework architecture and have different objectives. The MDA application utilizes message for the creation of feature requests using the MDF WebService to integrate corresponding features of the mobile device into a MDA workflow. The integration module MDI creates messages based on incoming events from external software systems such as PDM systems. The MDC uses different message types for the communication with the framework core in order to convey content such as sensor data but also feature states and performed user interaction results. The MDF uses different message types to interact with the corresponding framework components such as MDC, MDA, and MDI. A message can be derived from one of four possible message types. The message type `MobileFeatureRequest` and `MobileFeatureResponse` are used to request mobile features as well as return the feature state. Occurring

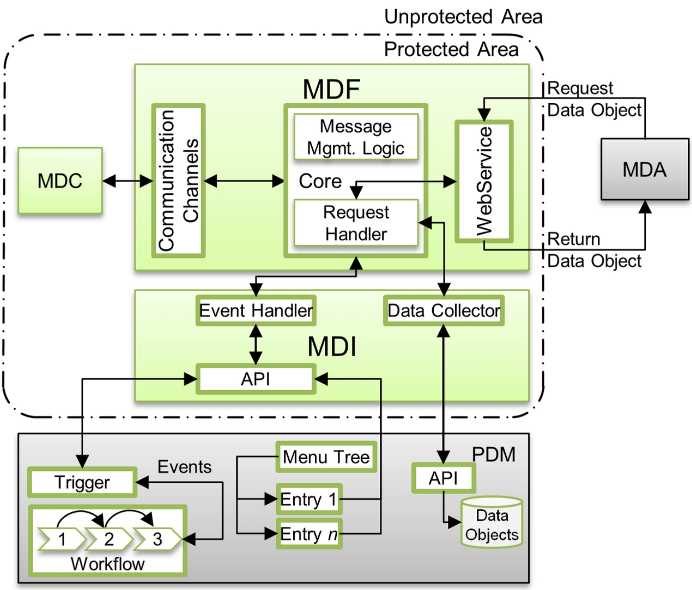


Figure 4.15: System integration model

events triggered in external software systems create a corresponding `Notification` to transport the information to mobile devices as well as coordinate the communicate channel with the framework. The message `Data` is only used in the mobile device for the transmission of continuous data streams. Each message is always assigned to a policy, which represents a set of rule for messages. For example, the rules set includes information about the maximum message size and priority. In addition, a single message is always assigned to exactly one clearly identifiable destination (e.g. mobile device) by the `MessageId`. A queue can have several messages whereas a message is always assigned to exactly one queue (see Figure 4.16). The message `MobileFeatureRequest` and `MobileFeatureResponse` are generated and processed in a synchronous workflow. A delayed delivery of these messages types is not possible, because the message delivery must be performed immediately to keep the workflow in sync. The message type `Notification` has the particularity that messages of framework are temporarily stored if the delivery to the mobile device or the MDA application is not possible. When the mobile device becomes available, the framework will register the device presence and will attempt to deliver the message again.

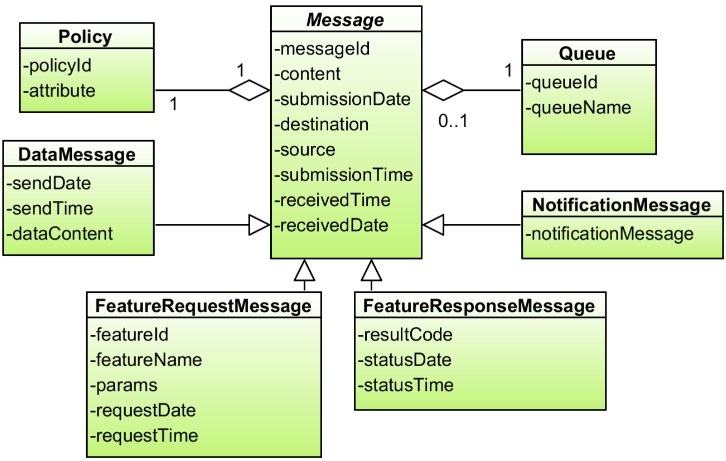


Figure 4.16: Message model - class diagram

In this case, the message delivery of type `Notification` can take place synchronous and asynchronous. The specification of the message types is done by the attributes. Messages of type `Data` are generally saved into a previously defined queue by the framework. In addition, these messages can either be transported to other message queue depending on the scenario or consumed by an application (e.g. MDA).

Communication Channels

The framework offers various communication channels for the message transport because not every mobile device supports all communication channels. The WebSocket protocol and the MQTT are provided as communication method with the framework. The MDC communicates to the framework with message type `Notification` to perform the device registration using the supported communication channel of the mobile device. Thereby, the MDC transmits some device-specific and environment specific information, such as the current time zone and device location. During the device registration, the framework defines the topic of communication channel exclusively for the mobile device. The topic on which the mobile device reacts as a receiver is composed of the Device-Id and Session-Id. The Session-Id is created previously by the authentication process of the framework using the username and password and is used for the dialogue with the MDC application. All further communication between the device and the framework takes place over the defined topic.

Message Transport Management

Once the device registration has been successfully performed, the framework becomes aware about the communication channel to communicate with the mobile device. The logic of the MDA application must determine the target device over the device properties to integrate a feature into the workflow of an MDA application. For this purpose, a query is sent to the framework's web service that returns the relevant information about the device. Subsequently, the MDA recognizes at this time the target device to create a web service message. This message contains the feature parameters and a callback address that are sent to the framework. The framework validates the incoming message from the MDA and puts the message on the corresponding communication channel of the mobile device. The mobile device receives the message, re-validates, and interprets the interaction to be performed. Once the feature interaction is finished with the user, the same communication channel is used to return the results to the framework. The framework validates and forwards the results to the MDA application using MDA's callback address (see Figure 4.17).

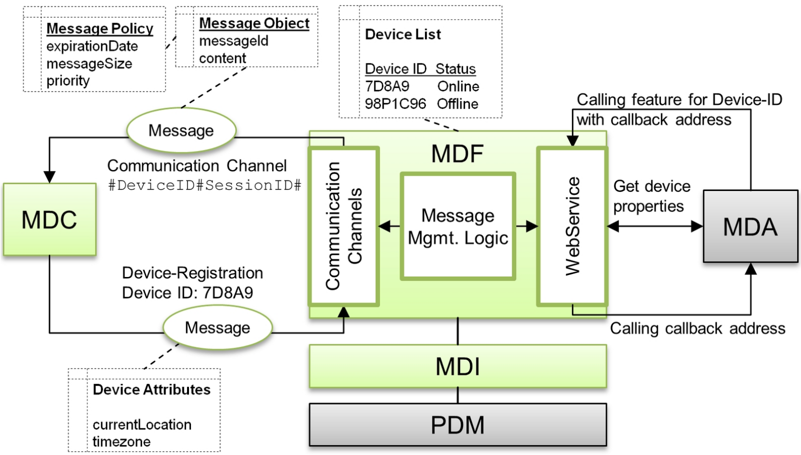


Figure 4.17: Message model workflow

Broker and Queue Management

Communication channels are managed and monitored by the broker instances within the framework. The broker has the responsibility to receive and send appropriate messages over the communication channel. Once the authentication process with the mobile device has successfully completed, the broker instance is initialized. Subsequently, the broker instance monitors the topic of the communication channel (based on the composed device ID and session ID). The broker instance terminates the monitoring of the topic when the session ID is invalid for certain reasons, e.g. user logs out (Session-logout), session expires (Session timeout). The broker instance switches at this time instantly into a passive mode (inactive). The passive mode of the broker instance allows the framework to efficiently initialize the topic monitoring without the need to generate a completely new broker instance which would cost additional time. The broker instance case has five lifecycle states, see Figure 4.18. Message flow in the framework and the message header are influenced by three policies. The policies define the next steps of the broker while handling the message. The device policy defines safety-relevant aspects and determines whether the client has the necessary authorization to send messages and the type of condition that must be fulfilled by the client to process a message by the framework. For this purpose, the message policy defines message-specific aspects. Therefore, the message size and the type of priority message are defined. For example, due to the synchronous communication to users in workflows, feature messages have a higher priority compared to notification messages. Queue policies are the third component that influences the message flow. Queue attributes for storing a large number of messages are taken into account at the place of destination. Depending on the message types and related message policy, the broker must execute the message processing and has to perform necessary process steps. While the message type specifies appropriate attributes in the header such as message destination, the message policy specifies the attributes such as maximum message size. In this context, message queues are mandatory for three reasons: (1) Delayed message delivery, (2) Data-driven message storage, (3) Overall message storage.

1. **Delayed message delivery:** Messages cannot always be delivered when the communication is in a faulty state. Therefore, a queue is needed to temporary store undelivered messages for later delivery.

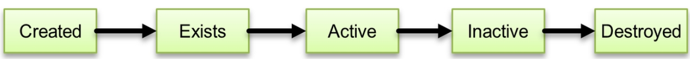


Figure 4.18: Message broker - lifecycle

Depending on the message type, messages are stored in a specially defined queue. The broker tries a new delivery of the message based on the defined rules in the message policy. As soon as a redelivery of the message has been executed successfully, the message is removed from the queue.

2. **Data-driven message storage:** Message type `Data`, which is created as an instance by the mobile device, needs a temporary queue as a buffer. The data message on the queue can either be forwarded to another queue or consumed by an MDA. Once a message is consumed, it will be removed from the queue.
3. **Overall message storage:** To trace the message flow in the framework, a copy of all messages is placed on a separate defined queue that has been processed by the framework. This establishes a high level of transparency to perform error analyzes and to ensure the auditability. Table 4.3 lists all framework components that create and receive corresponding message types.

Message Type	MDA		MDF		MDC		MDI	
	S	D	S	D	S	D	S	D
Data				x	x			
Notification	x					x	x	
FeatureRequest	x					x		
FeatureResponse		x			x			

Table 4.3: Message types by framework components (S=Source, D=Destination)

4.9.2 Security Model

The Security model of the framework covers all components (see Figure 4.19) and includes several safety characteristics. All MDA applications implement the session management of the framework and are registered on the framework. Therefore, an MDA application cannot be called directly, but the access is always performed over the framework. The framework manages all registered MDA applications and adds appropriate safety conditions. These safety conditions represent security features such as device ID, location of the mobile device, and username.

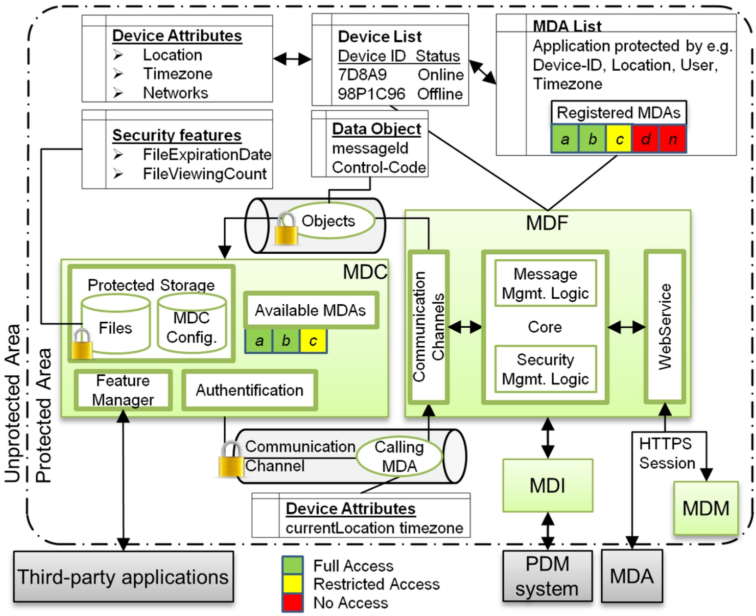


Figure 4.19: Security model

In case that a safety characteristic does not meet the defined criteria, the access to the MDA is blocked. This also applies for security characteristics, which cannot be collected from the mobile device. Therefore, other existing safety characteristics for the access protection must be used for such device types. In case that all safety characteristics for a MDA application are passed, the access to the MDA is granted and a list of MDA applications is returned to the mobile client MDC for visualization purposes. The MDC transmits the corresponding security characteristics to the framework during the initialization phase of the communication channel before any access to MDA applications is granted. The authentication process with the framework requires the username and password. All sensitive data are encrypted by the MDC and stored in a specially set up and secured data area. Confidential information, such as files containing 3D models and drawings can be marked by the MDA application with a security feature. A security feature may represent an expiration date or a maximum duration of a view file. Once a security feature exceeds the defined criteria, the MDC will perform the required steps to remove the content. For example, files which have reached an expiration time of the visibility would initiate a secure file deletion on the mobile device.

4.9.3 Data Privacy Model

The privacy model allows MDA applications and mobile users a maximum degree of flexibility with an adequate consideration of the privacy. The model is composed of three core elements. Since the privacy mainly applies to mobile users, the model is taken into account when considering functionality of self determination to protect user's privacy (see Figure 4.20). The core elements cover the following points:

1. Informational character: Usage of authorized mobile features requested by MDA applications to display them to the mobile users
2. User's prior consent for the usage of selected mobile features by MDA applications
3. User revokes the authorization of the usage for all or selected features in MDA applications

The internal flow takes place in the following steps:

1. During the registration process of the MDA application against the framework, the MDA application must provide information about used mobile features. The successful registration of the MDA application simultaneously represents the authorization to use defined features on the mobile device without the need to request further authorization from the mobile user. In case that a MDA application would request an additional mobile feature, which was not specified during the registration process of the MDA application, the framework would refuse the usage of this feature and returns an appropriate error message. Mobile features that have the appropriate usage authorization are processed by the framework and executed on the mobile device. When the first calling of the MDA application occurs, the mobile user obtains

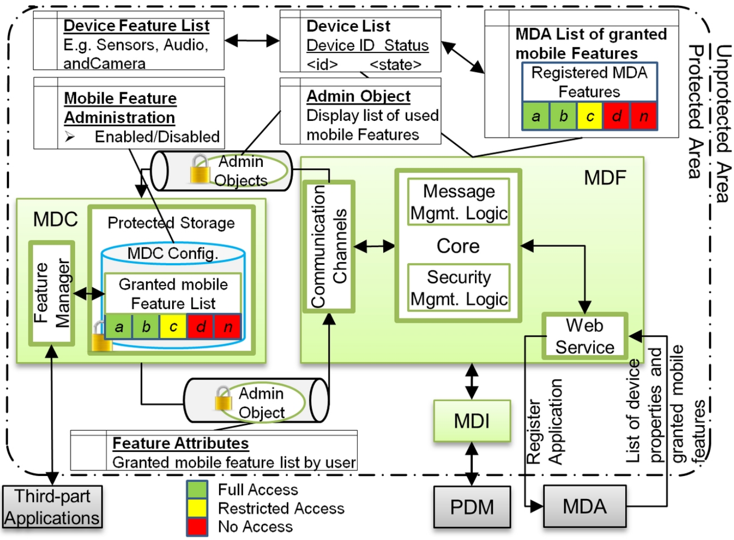


Figure 4.20: Privacy model

a list of mobile features displayed that may be used during the application execution.

2. When the first MDA application runs on startup, the mobile user receives a list of used mobile features displayed. In this case, the user must decide whether to accept or reject the feature usage. Once the user has issued its approval, the MDA application is executed in accordance with the definitions of the registration. In case that the mobile user refuses the usage of mobile features, the MDA decides whether the application can be executed or not. The MDA must provide to the mobile user an alternative workflow and must have it accordingly implemented. For example, the alternative for a voice-based search function could be a conventional text-based search function.
3. The user can influence the behavior of the MDC through the configuration and decides which mobile features be used in general. For this purpose, the user can specify the usage of mobile features for each MDA application individually. The configuration profile

of mobile features is subsequently transmitted and managed by the framework. The MDA application, which requests information from the framework, must decide whether the defined degree of functionality of the mobile user is sufficient to perform the functions in the MDA application. Otherwise, the MDA application needs to provide appropriate alternative functionalities or has to inform the mobile user that the function execution is not possible due to the reduced or restricted feature functionality.

4.9.4 User Directory Model

The Framework requires a methodology for authentication of mobile users when dealing with user accounts. Therefore, the framework must have a corresponding user model that reflects the processes between the mobile device, the destination system (e.g. PDM system), and the MDA application (see Figure 4.21). The framework can utilize an internal user management, but the usage of an external user directory has been prevailed over the years to authenticate and to ensure that only authorized users can utilize nearly all applications. In this case, the framework supports the authentication of users through the Lightweight Directory Access Protocol (LDAP). All framework components such as MDC, MDA, and MDM send the login data (username and password) to the MDF for the authentication. Subsequently, the MDF performs the authentication against the user directory (local database or LDAP). If the login is successful, a login token for the session is created. This login token is valid until the session has been terminated by the user or the session has expired. The login token consists of an alpha-numeric hash value, which is composed of a 16-byte hexadecimal number (UUID), a timestamp, the device ID, and the encryption of the entire hash. MDC and MDA can use a corresponding SSO service for the authentication to perform a one-time authentication for all services of the system. In this context, the selection and usage of appropriate SSO service providers will not be discussed any further.

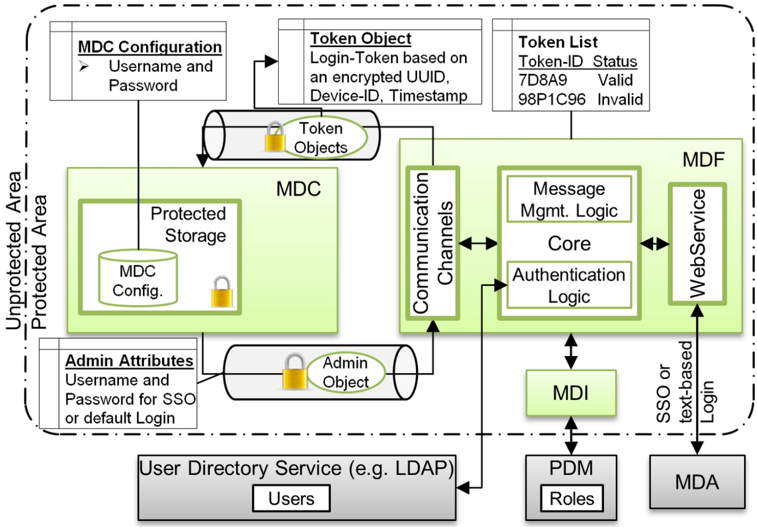


Figure 4.21: User model

4.10 Summary

This chapter has explained the objective, approach, and applied methodology. This was followed by the delimitation of contexts for the framework modeling. The use of ubiquitous PLM language ensured that the persons involved were using the same vocabulary. In addition, user stories were analyzed as well as common structures and correlations were recognized to identify framework patterns. Moreover, the corresponding patterns for the Mobile Feature Framework were derived based on the results of user stories and summarized in general groups. This was followed by the stepwise development of the overall domain model. The starting point was a rudimentary domain model, which was further designed and developed in the subsequent phases. Subsequently, the architecture was divided into multiple layers and corresponding components were designed. Furthermore, tasks and responsibilities were assigned to the components. Afterwards, objects and services were designed for these networked components and the behavior for certain conditions was specified. As a final and equally point, the development of cross-component models takes place for the purpose of generating the overall framework architecture.

The development of the domain model focused primarily on the conceptual design, rather than the realization of the components or the development and implementation of a feature concept. The challenge was to develop a flexible domain model that meets the requirements of users, systems, and various business aspects of Chapter 2. The domain model had thereby also reflected a special consideration of the technical and economic conditions as well as the openness to other software systems. All the necessary steps and aspects in the model building process of the overall model now form the basis for the ongoing design of a feature concept, which promotes the phases of planning, design, and construction of mobile features.

5 Feature Planning, Design, and Construction

The already modeled framework components must be considered for the planning, design, and construction of mobile features, which, therefore, can be used more efficiently. Firstly, the assignment of features into categories must be determined. For this purpose, feature categories must be created and its scope has to be described. Secondly, after creation, the features are assigned. All features are planned and implemented according to the *DeviceFeatureExtensibilityModel* (see Section 5.2). This model supports the design of the individual features through the central provision of basic functionalities. This prevents that each mobile feature uses separate connections for the communication and own code fragments, which result in a reduced effort of corrective maintenance. Finally, the lifecycle states and internal processes per feature are described in the phase of feature design and construction as well as discuss the technical implementations. In addition, the criteria definition for the inspection of the quality assurance takes place.

5.1 Determination of Feature Lists

Before mobile features can be planned, a group of possible features with various application purposes must be carried out. The feature grouping is used to identify features with same characteristics and to define the overriding objective. Therefore, the classification and definition of groups takes place in Section 5.1.1 and the categorization of features is carried out in Section 5.1.2.

5.1.1 Feature Groups

Feature groups can be classified according to functional aspects, technological aspects and regional-cultural aspects depending on the respective objective. Since the Mobile Feature Framework is focusing on the collaboration as well as contribution and consuming of information in context of PLM, the following four main categories are used for the grouping of features.

- **Mobile Data Collection (MDC):** This category contains all features that have the objective to capture information. The capturing can take place via sensors, camera, microphone, or device-specific data such as connection data of the mobile network or battery status.
- **Mobile Data Presentation (MDP):** This category contains features, which has the task to present information in different styles. Thereby, information can be visually presented and animated to support mobile users in the fulfillment of activities.
- **Interactive Collaboration (IC):** Data capturing and presentation are combined by a collaborative process. This category represents features following the common three-step: sequence of data collection, processing of the collected data, and presentation of results.
- **User Behavior and Environment (UBE):** The last category collects behavioral information from users and represents features that determine environment-related data of the mobile device. This allows mobile applications to react more granular in mobile contexts and to adapt against the user-specific behavior.

5.1.2 Categorization of the Features

Mobile features must be recorded in the scheme *‹Action›‹Result›‹Object›* to be categorized. Therefore, appropriate use cases following this scheme were created (see Appendix D.1). The listed features are derived from a fraction of these use cases, which were taken from the user stories in Section 4.4. Not all features can be planned and implemented, since this would go beyond the scope of this research work. The prioritization of mobile features depends primarily on the proportionality in terms of the scope of this work. Therefore, individual features are selected from each group in Table 5.1. The selected features are Number 2 and 3 from group MDC, Number 6 and 7 from group MDP and IC as well as Number 10 from group UBE, which are modeled and implemented in the further steps. The list is aimed at supporting the general approach adapted for this work and the many different aspects that should be considered as guidelines. Mobile features have been selected from each feature group, which varies in the complexity of design and implementation. In addition, features that have appropriate dependencies to third-party applications were selected to demonstrate the entire workflow (e.g. dependencies to providers such as barcode scan). All features are dependent on the respective mobile platforms, however, the focus is less on the mobile platform, and more on the ability to integrate various types of features into the Mobile Feature Framework. Therefore, the implementation of features using the Android platform is only an instrument to demonstrate the capabilities. Certainly, other mobile platforms can be used for the feature integration as well. The next section covers the modeling for the feature expansion of the framework component MDC.

5.2 Device Feature Extensibility Model

Today, features are diverse and can be found almost everywhere in mobile devices. However, all feature are currently not captured and taken into account by the feature abstraction due to additional hardware features and software enhancements promoted through new device models. In this context, the term feature-abstraction is understood as the generic use of hardware features such as camera, sensors, and microphone in combination with software features, which are mainly provided by third-party software providers. The features are managed and controlled by the MDC. During the initialization of the MDC, all defined feature modules are loaded dynamically, managed, and controlled by the MDC. Once a feature request message is received from the framework, the MDC analyzes and interprets the message. Subsequently, the MDC calls in context of the feature request the corresponding feature module and passes the control of a task for interaction with the user and control

No.	Business Activity Name	Feature List	Derived Pattern
Group: Mobile Data Collection (MDC)			
1	Speech Recognizer (Voice commands and gestures to interact with objects)	D.28	4.4.7
2	Object ID Identifier (Identification of real objects by ID codes)	D.29	4.4.2
3	Object Recognition Identifier (Identification of real objects by images)	D.30	4.4.2
4	Autonomous BOT (Autonomous transmission of data)	D.31	4.4.8
5	Media Contributor (Creation of media contributions in mobile contexts)	D.32	4.4.7
Group: Mobile Data Presentation (MDP)			
6	Environmental Object Presenter (Visualized virtual objects (e.g. map with objects in user's environment))	D.33	D.1.15
Group: Interactive Collaboration (IC)			
7	Speech Commander (Shaking perform specified user actions)	D.34	4.4.1
8	Object Assembler (Overlaid reality for the interaction of virtual and real objects)	D.40	4.4.5
9	Feature Authenticator (Validate user's fingerprint)	D.36	4.4.6
Group: User Behavior and Environment (UBE)			
10	User Location Examinator (Determine the location of the mobile user to consider the behavior and content in the application)	D.35	4.4.4
11	User Interrupter (User context consideration & control (e.g. low temperature changes the size of UI components))	D.37	4.4.1
12	User Environment Controller (Adjusts the volume based on the environmental noise)	D.38	4.4.1
13	User Brightness Examinator (Depending on the brightness of the mobile device, modify the colors of the web-based UI components)	D.39	4.4.1

Table 5.1: Feature list

purposes. Once the process of interaction is completed, a response feature message is created by the feature module and transmitted by the MDC to the framework using the communication channel. New feature modules can be flexibly integrated into the MDC. For this purpose, the feature module must comply with the defined structure which regulates the function call, the communication, and interaction with the user. The implementation of a feature is explained in Section 5.3. The device feature extension model is composed of a series of platform-specific components (see Figure 5.1). The validation of mobile users against the framework is accomplished by authentication modules. Once authentication has been performed, the communication component establishes the communication links as well as coordinates the exchange of messages over defined communication channels. The WebView of the MDC integrates the native browser of the mobile platform for web-based interaction with the framework. The most important component in this model is represented by the Feature Manager, which controls

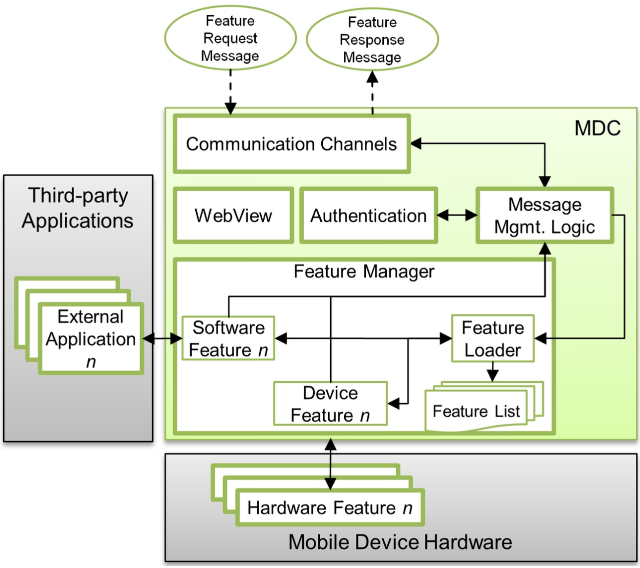


Figure 5.1: Device feature extension model

the individual feature modules for software and hardware together. All incoming feature messages are processed by the message logic as well as feature requests are forwarded to the Feature Manager. Subsequently, the feature manager calls the initialized feature modules and delivers the control to the feature module for further interaction. The feature modules can then perform further interaction with native hardware features or software provided by third-party providers. Once feature modules have completed the interaction with the user, the results are returned to the Feature Manager in order to create a feature response message, which is sent over the communication channel back to the framework.

5.3 Feature Design and Engineering

This section covers the process planning for selected features. All features are registered using a configuration file and loaded by the Plugin Manager. Once a message has been received, it is evaluated by the MDC. If the message contains a feature request, the Plugin Manager will be informed accordingly. Subsequently, the Plugin Manager determines the corresponding registered plugin. Once the Plugin Manager has determined the requested plugin with the related state, the plugin will be loaded and the message is passed on to the plugin. From this point, the plugin takes over the responsibility for all further interactions that are necessary to perform tasks. The plugin determines all transmitted message parameters using the basic functions of the MDC. Message parameters are essential during the runtime of the feature plugins. The message format consists of two parts *HEADER* and *DATA* (see Figure 5.2). The message header contains as mandatory declaration only the parameters Feature-Name and Feature-Action. In addition, the header of MobileFeature messages can optional contain parameters such as sender information of the MDA application, timestamp of the message creation as well as system information about the MDF in context of the message (e.g. checksum of feature message). On the one hand, the header contains no information about the technical delivery of *MobileFeature* messages and is only needed by the MDC. On the other hand, it is not used by the Feature Plugin. The FeaturePlugin obtains the necessary data from the *DATA* field. This field contains a HEADER and DATA section.

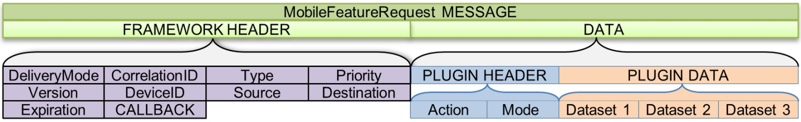


Figure 5.2: Message protocol for a feature request

However, both sections are reserved exclusively for the FeaturePlugin. The HEADER of the FeaturePlugin includes mandatory information such as ENCODING type of DATA field as well as the CALLBACK-URL. The ENCODING parameter specifies the used character set and character encoding. The parameter CALLBACK-URL contains the defined URL to call the MDA application under specified conditions after the completion of the FeaturePlugin. This procedure matches with the design pattern of the Inversion of Control (IoC), which mostly reflects common workflow methods of frameworks. FeaturePlugin always consists of basic fields and feature-specific fields. The basic fields include the *EntryPoint*, *Core*, *Third-Party-Activity*, *Helpers*, and *Communication*. The EntryPoint is the starting point for the FeaturePlugin after the feature call of the PluginManager. All the capabilities are organized and structured in context of the FeaturePlugin, for example performing interactions between the plug-in and the mobile user. The EntryPoint is connected to the core of the FeaturePlugins, which contains all core functionalities of the feature. Since the FeaturePlugin cannot perform all tasks, there is a dedicated field for third-party activity to delegate tasks to other mobile applications. The Helpers of the FeaturePlugins provide help functionalities that cannot be fulfilled by the basic functions of the MDC. The communication area of the feature is used to retrieve additional data over a communication link and perform the callback function of the plugin to the MDA application.

Featurization of webpages

In order to use mobile features in the user context, the webpage called by the user needs to be adapted. For this purpose, a featurization of the webpage is necessary in order to allow the mobile feature to operate in the user context. The term *featurization* was applied by the author of this research work and used for the Mobile Feature Framework to describe the adaptation and revaluation of MDA webpages with mobile feature. The use of mobile features would not be possible in the user context without applying the mandatory featurization steps for webpages. The featurization of webpages takes place in four steps:

1. Definition of valid actions for the webpage (e.g. lifecycle webpage: Object state of document DOC-0001 can be promoted and demoted)
2. Assignment of the device-driven feature request to a suitable mobile feature of the webpage (e.g. speech recognition, gesture recognition, or facial expression recognition)
3. Storing of recently accessed data in the user session
4. Interpretation and assignment of the feature result to the defined object action

In order to call and process the device-driven feature request within the user context, a synchronization of the sessions between the feature context and the user context takes place through the MDC. Code Example 5.3 shows schematically the code structures used for the MDA application in a very simplified form.

5.3.1 Selected Feature: Object ID Identifier

The objective of the ObjectID identifier features is to enable the identification of objects accessed by ID-code using the abstracted pattern D.1.8 in order to allow MDA applications to perform further actions based on the identified object. Mobile users are using their mobile device to carry out the identification of objects. After the detection of the unique ID code, the user receives the corresponding results presented. The feature has the tasks to act as a barcode reader to scan bar codes. Various barcode types are read and passed on to the MDC to transmit the data to the MDF. The recognition of barcodes takes place through the integrated digital camera of the mobile device. Hardware-based barcode readers usually perform the bar code detection using red and infrared light. The captured image is processed and evaluated by a provided library. Subsequently, the determined barcode type and barcode string are returned by the MDC to the MDF. The barcode string is represented by an alphanumeric string and its length character, which depends on the barcode type. Barcode types

```
/*
 * Called by the mobile device
 */
public void deviceDrivenfeatureRequestDispatcher () {
  // Choose mobile feature for actual webpage displayed to the user
  // ...

  // Set callback the last page visited by the mobile user
  featureOptions.put("callbackAddress", url);
  featureOptions.put("userContext", data);

  // Call mobile feature
  webService.invokeMobileFeature(FeatureName);
}

/*
 * Called by the mobile user
 */
public String showLifecycleIssueObject() {
  // Featurization of this page
  if (request.getParameter("deviceDrivenfeatureAction").equals("yes")) {
    // Analysis feature results
    // ...

    if (results.contains("promote")) {
      // Perform promote action of current object
      webService.promoteObject(userSession.getAttribute(name));
    } else {
      // Other object actions ...
    }
  } else {
    // Store object information into session
    userSession.setAttribute(name);
  }
  // Loading object data..
  webService.getObjectDetails(name);
  // Load page template and send output to user
  templateManager.load(templateName);
}
```

Figure 5.3: Webpage featurization for MDA applications

such as EAN-13[1] and UPC-A[2] also include a check digit to early recognize reading errors caused by difficult lighting conditions, damaged codes as well as codes that have been attached on moving mechanical components. The MDF determines the callback address of the MDA application and forwards the results of the barcode scanning process to the MDA application. The MDA can now process the data (e.g. validation) and initiate further interactions with the user. The requirement for the mobile device is an integrated or external camera that can be controlled by the mobile platform. The object identification is performed by server-side components and thus no additional requirements are addressed to the device. A library for image processing and image decoding is necessary to analyze and determine the barcode embedded in the captured image. For this purpose, there is a wide range of existing libraries (e.g. ZBar[3], Scandit's Barcode Scanner SDK[4], and shopsavvy's Barcode Scanner SDK[5]). For the feature implementation of the barcode scanner, the open-source library ZXing[6] was selected and is implemented in the Java programming language. The library supports the detection of various 1D products (e.g. EAN-13) and 1D industrial codes (e.g. Code 128) as well as 2D codes (e.g. QR Code). Since the extensive library is not part of the MDC core and is fully considered as an external library, the scanning process with related libraries is outsourced in a separate library project. The feature interacts with the library and obtains the returned results as soon as the scanning is completed. The sequence

[1] European Article Number (EAN) is the former name for the Global Trade Item Number (GTIN) for unique identification of products. The number 13 represents the count of digits of the code.

[2] Universal Product Code (UPC) was introduced in 1973 in the USA and is used for the identification of retail products by a bar code[Cf. Zsi12, p. 280]. The EAN is compatible to the UPC.

[3] ZBar bar code reader is an open source software suite for various bar codes types and online available at http://zbar.sourceforge.net/

[4] Scandit's Barcode Scanner SDK is a commercial software library working on various mobile platforms and is online available at http://www.scandit.com/products/barcode-scanner/

[5] Scandit's Barcode Scanner SDK is a commercial software library, includes product details for more than 20,000,000 products and is online available at http://shopsavvy.com/developers

[6] ZXing (zebra crossing) is an open source software library developed for Android and is online available at https://github.com/zxing/zxing/

diagram in Figure 5.4 illustrates the consecution between the MDC-Core, feature, and the barcode library. The body class must implement the predefined methods in order to provide a common feature structure for all mobile features. These methods are called, respectively, `execute()` to perform the feature action and `onActivityResult()` to process the results of the feature. The method `execute()` analyzes the feature request for the BarcodeScan and performs the mapping of the framework parameters (e.g. callback address of the framework) and the specific BarcodeScan parameters (e.g. feature action, accepted type of codes). Subsequently, the `execute()` method delegates the scanning request to the method `scan()`, which initiates the interaction of the scanning process with the third-party library. Once the scanning process has been completed, the method `onActivityResult()` is called to return the results. This method now creates a feature-response message with the corresponding results (Identified barcode ID and barcode format) and

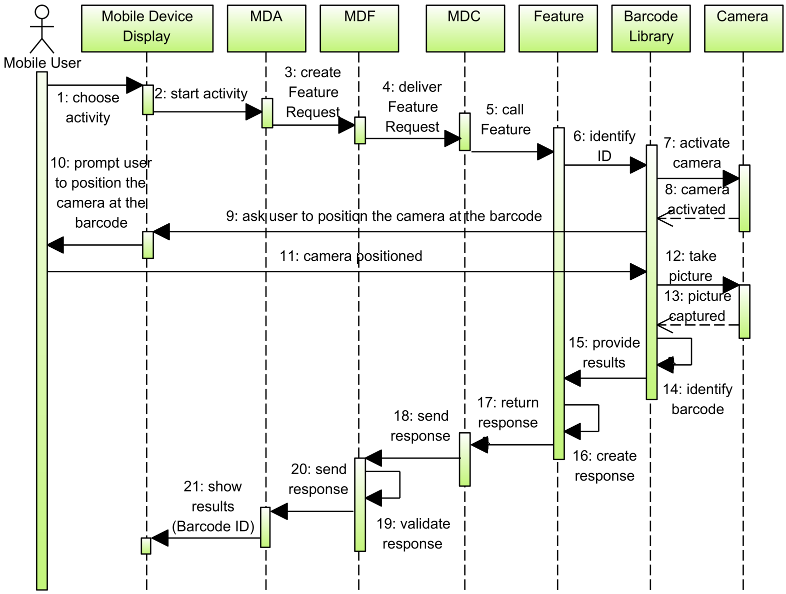


Figure 5.4: Sequence diagram for feature Barcode Scan

uses the corresponding message delivery method of the MDC to transfer the message to the MDF. Thus, the feature is realized in the MDC, so that the MDA application can use the BarcodeScan feature. The MDA must collect all specific parameters of the BarcodeScan feature (e.g. callback address of the MDA application) to submit the request to the web service client. The generic web service client of the MDA application transmits the request with the parameters to the MDF, which forwards the request to the mobile device.

```
wsc.invokeMobileFeature(applicationId, SSOToken,
serviceAction, featureName, featureAction, options);
```

Once the MDC calls the callback address of the framework to return the results to the MDA application, the transmitted Session-Id of the MDA is verified by the MDF. In case that the MDA session is invalid, the results of the barcode scan are not passed to the MDA application and the callback request is redirected to the login page. This process is one aspect of the security architecture of the framework. In case that the session is valid, the callback to the MDA application is performed by the framework, which reads the defined MDA callback address of the feature request. Subsequently, the results can be extracted and further processed by the MDA application.

5.3.2 Selected Feature: Object Recognition Identifier

This feature is derived from the abstract pattern D.1.7. The objective is the identification of objects with the camera of the mobile device. With the identified object, the MDA application should initiate further actions. For this purpose, the mobile user has to capture the whole real object with the camera and receives returned data results of the identified object. This feature is utilized for object recognition using the integrated digital camera of the mobile device. The MDA application requests the feature to create an image of the real object. Subsequently, the captured image is transmitted to the server for image recognition. The image recognition attempts to segment the object in the photo and to assign the previously defined description to the object. However, there is no detection of interactions between multiple objects, such as the interaction between brake, harness, and taillight of a vehicle. The name of the object is returned as result of the image recognition. Based on the object name, further information can be retrieved from the PDM system. Mathematical information and attributes about an object must be available for recognition. Such attributes are, for example, shape, texture, size, number of edges, and color of the object. This kind of data attributes with the object description are stored into a database. This information may be available in the PDM system, so that no new data acquisition is necessary. In this case, the required information can be retrieved through the framework component MDI. Mobile device must have a built-in or external camera, which can be controlled by the mobile platform for the development of this feature. The object identification by image recognition is performed by server-side components, so that no additional requirements are addressed to the device. The fact that the object identification by image recognition has a high degree of complexity and further server-side components are required (e.g. image database, and image indexing), the entire feature cannot be realized by a single library on the mobile device.

The server-side required components for image recognition can be used either as a service from a third party provider or service from their own image recognition components. The body class must implement the predefined methods in order to provide a common feature structure for all mobile features. These methods are called, respectively, `execute()` to perform the feature action and `onActivityResult()` to process the results of the feature. The method `execute()` analyzes the feature request for the ObjectIdentifier and performs the mapping of the framework parameters (e.g. callback address of the framework) and the specific ObjectIdentifier parameters (e.g. feature action, API key, client Id). Subsequently, the `execute()` method delegates the recognize request to the method `recognize()`, which initiates the interaction with image recognition library of the third-party provider. Once the recognition process has been completed, the method `onActivityResult()` is called to return the results. This method now creates a feature-response message with the corresponding results (Identified barcode ID and recognition status) and uses the corresponding message delivery method of the MDC to transfer the message to the MDF. Thus, the feature is realized in the MDC, so that the MDA application can use the ObjectIdentifier feature. The MDA must collect all specific parameters of the ObjectIdentifier feature (e.g. callback address of the MDA application) to submit the request to the web service client. The generic web service client of the MDA application transmits the request with the parameters to the MDF, which forwards the request to the mobile device.

```
wsc.invokeMobileFeature(applicationId, SSOToken,
serviceAction, featureName, featureAction, options);
```

Once the MDC calls the callback address of the framework to return the results to the MDA application, the transmitted Session-Id of the MDA is verified by the MDF. In case that the MDA session is invalid, the results of the recogition are not passed to the MDA application and the callback request is redirected to the login page. This process is one aspect of the security architecture of the framework. In case that the session is valid, the callback to the MDA application is performed by the framework, which reads the defined MDA callback address of the feature request. Subsequently, the results can be extracted and further processed by the MDA application.

5.3.3 Selected Feature: Environmental Object Presenter

The Environmental Object Presenter feature has the objective to implement the abstracted pattern D.1.15 as basis for the visualization of objects on a map under consideration of the user context. This allows users to perceive an enhanced perception of PLM objects in the immediate environment. The used pattern describes the generic visualization of objects, but not the detailed concept and implementation of the visualization through maps. Therefore, this pattern can be used to generate various mobile features for different contexts. The MDA application determines the current location of the mobile user using the feature. The actual location data are used by the MDA application in order to retrieve corresponding PLM objects using the MDI component. The MDA application creates a separate data layer for each object type and combines the data layers into a data request. The data layers are structured as presented in Figure 5.5. Subsequently, the MDA application calls the feature with the corresponding data to represent the visualized information to the mobile user. The derived requirements of this feature are a built-in or external camera and a GPS receiver to localize the mobile user, which can be controlled by the mobile platform. In order to visualize information for mobile users, an interface for augmented reality software, maps service, or other visualization components are needed. For this purpose, a wide range of providers for AR software (e.g. Argon[7], Wikitude[8], Layar[9]) and mapping services (e.g. OpenStreetMap[10], Nokia HERE[11], Google Maps[12], Bing Maps[13]) exist. The AR software Wikitude has been selected for the feature realization, which provides an interface for the mobile platform Android. Since the AR software Wikitude is not part of the MDC core and is considered as an external third-party solution, the visualization process with related libraries are outsourced in a separate library project. The feature interacts with this library and receives returned the results after completion of the visualization. The body class must implement the predefined methods in order to provide a common feature structure for all mobile features. These methods are called, respectively, `execute()` to perform the feature action and `onActivityResult()` to process the results of the feature. The method `execute()` analyzes the feature request and performs the mapping of the framework parameters (e.g. callback address of the framework) and the specific feature parameters (e.g. feature action, data layers). Subsequently, the `execute()` method delegates the visualization request to the method `callThirdPartyEngine()` which initiates the interaction with the visualization component of the third-party provider.

[7] Argon is an open source augmented reality browser by the GVU research center affiliated with the Georgia Institute of Technology

[8] Wikitude is a proprietary AR library for mobile platforms using web technologies.

[9] Layar is an AR browser as mobile application to find various objects

[10] OpenStreetMap (OSM) is a collaborative mapping involves web maps and user-generated content (UGC) and was developed in response to the supremacy of proprietary map services such as Google Maps.

[11] Nokia HERE formerly Ovi Maps is a cloud-based map model which provides features such as Point of Interest (POI), pedestrian navigation, and speed warnings independent of the device type.

[12] Google Maps is a map service that provides feature such as personalized maps, POI as well as location-based and subject-based maps.

[13] Bing Maps formerly Windows Live Maps is a map service by Microsoft.

The data contain any number of layers, while a layer may represent a particular type of object with additional visualization properties. The user is able to explore and inspect objects of different types in the various visualization layers. Once the user has completed the examination of objects in the visual environment, the method `onActivityResult()` is called to return the results. Finally, the callback of MDA application takes place using the previously defined callback address in the feature request header. Therefore, the method `onActivityResult()` creates a feature-response message with the corresponding results and uses the corresponding message delivery method of the MDC to transfer the message to the MDF. The feature logic for the visualization is realized in the MDC, so that the MDA application can use the feature using the framework. The MDA application has to determine the location data of the mobile user using a separate feature. Based on the localization data of the feature, the MDA application requests the objects over the integration component MDI from the PDM system. This objects are located in the surroundings of user's determined location. For this purpose, the data model of the PDM system must be supplemented by additional attributes in regards to the position of objects. The MDA application collects all feature-specific parameters (e.g. callback address of the framework, various data layers and layer properties) to invoke the web service client. The generic web service client of the MDA application submits the request with the parameters to the MDF, which subsequently forwards the request to the mobile device. Once the MDC calls the callback address of the framework to return the results to the MDA application, the transmitted Session-Id of the MDA is verified by the MDF. In case that the MDA session is invalid, the results of the visualization are not passed to the MDA application and the callback request is redirected to the login page. This process is one aspect of the security architecture of the framework. In case that the session is valid, the callback to the MDA application is performed by the framework, which reads the defined MDA callback address of the feature request. Subsequently, the results can be extracted and further processed by the MDA application.

```
JSONObject overlay0Data0 = new JSONObject();
JSONObject overlay0Data1 = new JSONObject();
JSONObject overlay1Data0 = new JSONObject();
JSONObject overlay1Data1 = new JSONObject();
JSONArray overlay1allData = new JSONArray();
JSONArray allData = new JSONArray();
JSONObject options = new JSONObject();
JSONArray totalList = new JSONArray();

options.put("animatePoint", "0");
options.put("zoomLevel", "3");

// First overlay: Visualize Team 1
overlay0Data0.put("latitude", "50.000000"); // Mainz, Germany
overlay0Data0.put("longitude", "8.270000");
overlay0Data0.put("name", "Test Designer");
overlay0Data0.put("description", "Designing electrical parts");

overlay0Data1.put("latitude", "48.8593"); // Paris, France
overlay0Data1.put("longitude", "2.3524");
overlay0Data1.put("name", "Test Reviewer");
overlay0Data1.put("description", "Review electrical parts");

// Second overlay: Visualize Team 2
overlay1Data0.put("latitude", "37.63"); // San Francisco, California
overlay1Data0.put("longitude", "-122.424");
overlay1Data0.put("name", "Test ProjectLead");
overlay1Data0.put("description", "Project Leader");

overlay1Data1.put("latitude", "35.68"); // Tokyo, Japan
overlay1Data1.put("longitude", "139.76");
overlay1Data1.put("name", "Test ManufacturingEngineer");
overlay1Data1.put("description", "Manufacturing employee");

overlay0allData.put(overlay0Data0);
overlay0allData.put(overlay0Data1);
overlay1allData.put(overlay1Data0);
overlay1allData.put(overlay1Data1);

allData.put(overlay0allData);
allData.put(overlay1allData);

totalList.put(allData);
totalList.put(options);
```

Figure 5.5: Data layers processed by the mobile feature

5.3.4 Selected Feature: Speech Commander

The objective of this feature is to implement the mapping between voice and actions based on the abstracted pattern D.1.3. This gives the user the possibility to fill in form fields and call frequently used functions by voice commands as it is the case for speed dial feature of common telephones. The user shakes the mobile device, which triggers an event that is recognized by the vibration listener (see Figure 5.6). The listener delegates the event to the Event Controller, which analyzes and assigns the recognized vibration as well as informs the Message Controller in order to create and send a mobile feature request to the framework. The triggered feature requested by the mobile device is used to inquire a suitable mobile feature for the current user context. For this purpose, the Shake event is only used as a method to trigger the feature request in order to inform the mobile application that a mobile feature for the actual context (e.g. webpage) is used. Based on the user context (e.g. lifecycle page of an object), the mobile application (MDA) sends an appropriate feature request to the mobile device to load the desired feature. In this feature case, the SpeechCommander is activated by the feature request. The SpeechCommander feature operates in context of the user session. Moreover, the webpage requires the featurization of the webpage which was already explained in Section 5.3. The Speech Commander feature enables the voice recognition on the mobile device and returns a list of possible terms as well as transmits the prepared results back to the featurized webpage. Subsequently, the webpage performs the analysis and evaluation of the proposed terms by using the mapping table. Once a phrase has been identified by the feature, the pre-defined action is initiated for the users. In case that the proposed term does not exist in the assignment table or the voice recognition software could not provide a list of proposed terms, a corresponding acoustic and visual warning is presented to the user. The derived requirements of this feature are a microphone and an integrated speech recognition software that can be controlled by the mobile platform. In addition, the spoken language of the user must be supported by the speech recognition software. Mobile platforms offer integrated speech recognition components. Google Android provides the voice recognition software Google Now, whereas

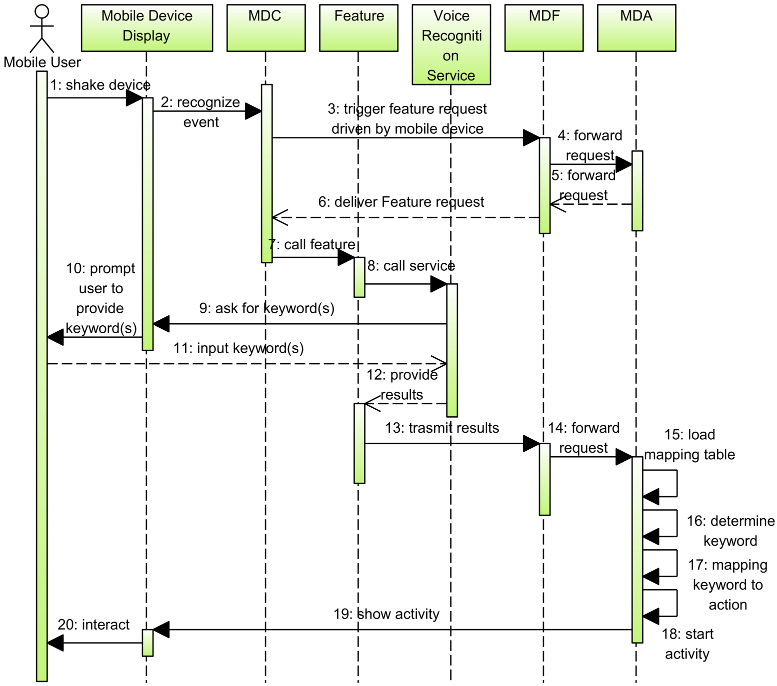


Figure 5.6: Sequence diagram for feature Speech Commander

Apple offers Siri for iOS. However, also third-party providers such as Maluuba, Iris by Dexetra and S Voice by Samsung can be integrated for this purpose. The body class must implement the predefined methods in order to provide a common feature structure for all mobile features. These methods are called, respectively, `execute()` to perform the feature action and `onActivityResult()` to process the results of the feature. The method `execute()` analyzes the feature request for the SpeechCommander and performs the mapping of the feature-specific parameters (e.g. count of proposed terms). Subsequently, the `execute()` method delegates the request to the method `callSpeechEngine()`, which initiates the interaction of the speech recognition software. Once the user has articulated the word phrase to the speech recognition module, the method `onActivityResult()` is called to return a list of the results which contains the proposed terms. Subsequently, the `downloadMappingTable()` method is called and downloads the user-specific mapping table. After that, the method `mapSpeechResults()` is called to perform the mapping between the terms and related actions. If none of the returned items matches with the assignment table, the method `noTermFound()` is called to inform the user about the result. In case of a positive match, the previously defined user action is executed by the `callUserAction()` method. The logic of the feature is implemented within the MDC, so that only the mapping table must be implemented by server-side components. The MDC calls the feature directly, so that no further communication to the MDF is necessary. A valid session with the MDF must exist to perform user-specific actions. If no valid session exists, only general actions (e.g. program exit, calling the browser) can be performed by the user. This procedure represents one aspect of the security architecture of the framework.

5.3.5 Selected Feature: User Location Examinator

The User Location Examinator feature has the objective to realize the abstracted pattern D.1.12 as basis for the localization of mobile users. This simplifies the user's perception and interaction with objects in his immediate environment. The user has no contact points with this feature, because the determination of user's location is performed as background process without any interactions with the user. The feature is called by the MDA application through the web service client, which forwards the request through the MDF to the mobile device. The feature determines user's location in the background and returns the results through the MDF to the MDA. The MDA is now able to use the localization data for specific activities and tasks (e.g. web-based visualization of objects using a map service). The derived requirements of this feature are a GPS receiver and a wireless module for faster position tracking, which can be controlled by the mobile platform. This requirement can be optional. Usually there are no dependencies to external libraries, because the query of localization information is an integral part of mobile platforms. The body class must implement the predefined methods in order to provide a common feature structure for all mobile features. These methods are called, respectively, `execute()` to perform the feature action and `onActivityResult()` to process the results of the feature. The method `execute()` analyzes the feature request and performs the mapping of the framework parameters (e.g. callback address of the framework) and the specific parameters (e.g. feature action). Subsequently, the `execute()` method delegates the request to the method `getLocationData()`, which initiates the platform-specific API calls to retrieve the location data. Once the localization data are available, the method `onActivityResult()` is called to handle the results. This method now creates a feature-response message with the corresponding results (for example, longitude, latitude, providers, and network name) and uses the corresponding message delivery method of the MDC to transfer the message to the MDF. Thus, the feature is realized in the MDC, so that the MDA application can use the localization feature. The MDA application must collect all specific parameters of the feature (e.g. callback address of the MDA application) to submit the request to the web service client. The generic web service client of the MDA application transmits the request with the parameters to the MDF, which forwards the request to the mobile device. Once the MDC calls the callback address of the framework to return the localization results to the MDA application, the transmitted Session-Id of the MDA application is verified by the MDF. In case that the MDA session is invalid, the localization data of the feature are not passed to the MDA application and the callback request is forwarded to the login page. This process is one aspect of the security architecture of the framework. In case that the session is valid, the callback to the MDA application is performed by the framework, which reads the defined MDA callback address of the feature request. Subsequently, the results can be extracted and further processed by the MDA application.

5.4 Summary

In the first instance, feature groups were created, defined and subsequently were assigned to the corresponding categories. In this case, they can be adapted to serve logical grouping of purposes for mobile features have been used to identify distinguishing and similar characteristics. Afterwards the Device Feature Extensibility Model was developed to provide a supportive frame for communication, interaction and control to the mobile features. Consequently, message protocol was deeply discussed to trigger mobile features.

In context of the Device Feature Extensibility Model, selected mobile features were designed and implemented. Each mobile feature showed other challenges that were faced during the conception and realization phase. In summary it can be stated that the feature model not only provides a frame for the creation of various mobile features, but also supports the communication flow between mobile features and server-side framework components as part of a piece of the mobile framework component. The provided interfaces of this model allow mobile features to reuse existing communication channels and message formats, instead of implement their own communication structures. In addition, safety-relevant aspects such as user authentication were covered. On the basis of this model, mobile features can be integrated efficiently in the framework and allow generic reuse for any desired applications.

6 Framework Engineering

This chapter deals with the implementation of the conceptual domain model in a representative language. A multitude of features for various scenarios was designed and implemented using the framework components previously in Chapter four. The following sections describe the environment and project structure of the framework engineering as well as framework components that have been implemented in various development projects as a prototype. This chapter focuses more specifically on the principal approach and less on the detailed description of all program-technical challenges, class structures and hierarchies as well as guidelines for the general naming convention. Section 6.1 covers the conceptual components of the framework in a landscape of existing middleware components. Section 6.2 describes the design and implementation of the project structure. Subsequently, Section 6.3 and 6.4 present the framework structure and discuss the database model which is essential for the implementation of the object persistence. Sections 6.5 and 6.6 comprise the integration of external user directories into the framework as well as infrastructure components that are used by the framework. The final Section 6.7 deals with the implementation of the framework testing.

6.1 Environment

An integrated development environment was necessary and has been defined that met the requirements for the implementation in order to permit the realization of the prototype. For this purpose, some decisions have been made regarding required components, that represent the composite development environment (see Figure 6.1). The realization of the prototype is not limited to the selected components, so that the realization with alternative components would be equally possible. Present discussions justify the decisions about the usage of certain components. The object-oriented programming language Java has been selected for the implementation of server-side components, because the language offers some basic advantages such as platform independence, ability of multithreading, memory management as well as strongly typed language compared to other programming languages. Eclipse was chosen as an integrated development environment (IDE), because this comprehensive development platform places a strong focus on creation of integrated web and application tools and is already established for use in the research and commercial environment. Alternative IDEs such as NetBeans and IBM Rational Application Developer (RAD) are comparable in efficiency with Eclipse. The Android mobile platform has been selected for the integration of mobile features in the PDM system, since it is supported by a variety of hardware manufacturers for smartphones, tablets and wearable devices. In addition, the Android platform shows the highest market penetration for smartphones at 81.3 percent [Cf. Fin13]. From

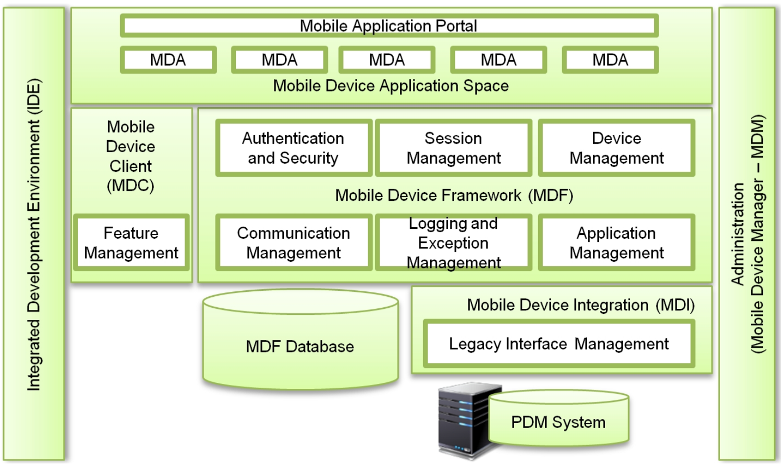


Figure 6.1: Environment components

a technical standpoint, mobile platforms like iOS and Windows Phone can be used as an alternative component as well, but the current market penetration is at a low level. Smartphones and tablets have been used as mobile devices that support the Android mobile platform and have hardware features such as camera and sensors already integrated. The WebSocket protocol and MQTT were selected for the communication between mobile devices and server-side components, because they meet the defined requirements that have been described in Section 2.3.2. The WebSocket protocol provides a full-duplex communications channel using the TCP protocol and has a lower communication overhead compared to a HTTP connection, because the HTTP header must be sent again for each new connection. MQTT has its strengths in the M2M communications[1] for automated data exchange between mobile devices, which are of importance in context of cyber-physical systems.

6.2 Project Structure

Due to the framework modeling in components and the resulting responsibilities (described in Section 4.7), the project structure has been aligned accordingly. Thus, the contexts of the entire domain model can be divided and various aspects can be treated separately. Therefore, the assignment takes place under the scheme: A context corresponds to a specific data source and/or project. The projects are mapped to the defined contexts from Section 4.2 as follows in Table 6.1. The implementation of the projects MDF and MDA takes place through dynamic web projects in Eclipse, because dynamic Java EE components such as servlets and JSP as well as static components (file types like image and HTML) are required. In addition, these types of projects fulfill all necessary basic requirements such as project structure and project configuration as well as provide various project perspectives for the respective development tasks. The MDC project is implemented for the Android mobile platform through an Android project in Eclipse, which is solely run on this platform. For this purpose, the Android SDK is required to provide the necessary API libraries and development tools

[1] Machine-to-Machine (M2M)

for creating, testing, and debugging Android applications. The realization of the MDC on other mobile platforms requires different IDEs and SDKs, which are provided by manufacturers. Thus, the implementation of the MDC on the mobile platform iOS requires Xcode development environment, whereas the Windows Phone requires the Visual Studio with the respective SDKs. The PDM integration component MDI requires two projects. The MDF-related MDI component is responsible for the processing and coordination of communications between MDF and the PDM system. This project is realized by a dynamic web project in Eclipse IDE. The second project provides access to the PDM system through a web service. This PDM-related MDI component does not include any business logic and acts only as a gateway between the framework and the data repository of the PDM system. This project requires a Java project that includes the proprietary API libraries of the PDM system. The generated byte code of this project is integrated into the code structure of the PDM system components. This integration procedure is commonly known as PDM customization.

Context	Project Name and Description
Mobile features	MDC (Mobile Device Client): Interaction client between user and framework
Core context	MDF (Mobile Device Framework): Coordination of communication between framework components
PLM	MDI (Mobile Device Integration for PDM): Integrator for PDM systems
Web applications	MDA (Mobile Device Application): Implementation of PLM applications using mobile features

Table 6.1: Assignment of contexts to projects

6.3 Framework Structure

Each of the projects listed in Table 6.1 has its own package structure. The package structure of the projects has been implemented according to the Domain Driven Design. The source code that represents the respective components of the framework, was classified in the package structure according to the logical architecture layers. Thus, the base package for the domain is `edu.kit.mdf.domain.model.*` (also called `[model]`). The source code in this package is independent from other packages. Accordingly, there is no dependence to the source code of the infrastructure package `edu.kit.mdf.infrastructure.*` (also called `[infra]`). In addition, each modeled aggregate of the domain has been assigned to a subpackage. Therefore, the subpackages `[model].mobiledevice`, `[model].application`, `[model].queue`, and `[model].broker` are under the base package `[model]`. The building blocks repositories and factories of the domain model are assigned to the same subpackage. Domain Services that do not serve a single aggregate are represented by another separately defined subpackage below, instead of within the existing subpackage.

6.4 Persistence of Objects

The realization of the persistence usually requires technology-specific implementations, which are reflected within the base package `[infra].persistence`. Below this package there are specific implementations for the object-relational mapping (ORM[2]) reflected by subpackage `[infra].persistence.hibernate`. The objects are distributed on several database tables in order to transform the relations in the normal form. Likewise, the distributed datasets that represent the object are reassembled by database operators (e.g. join and union) within the transaction. The data of the MDF are stored in the relational database management system MySQL and administrated by the user interface phpMyAdmin. The administration includes the definition, modification, and deletion of the database and table structures with the associated field definitions for the implementation of the relational database model. The database model of the framework describes the technical implementation of the object data in the DBMS from the perspective of the infrastructure layer. Figure 6.2 illustrates the database model of the framework. The

[2] Object-Relational Mapping (ORM) is used for mapping of objects in relational databases

database model consists of 20 tables that are connected to each other. Table 6.2 lists the tables with a brief description of the purpose.

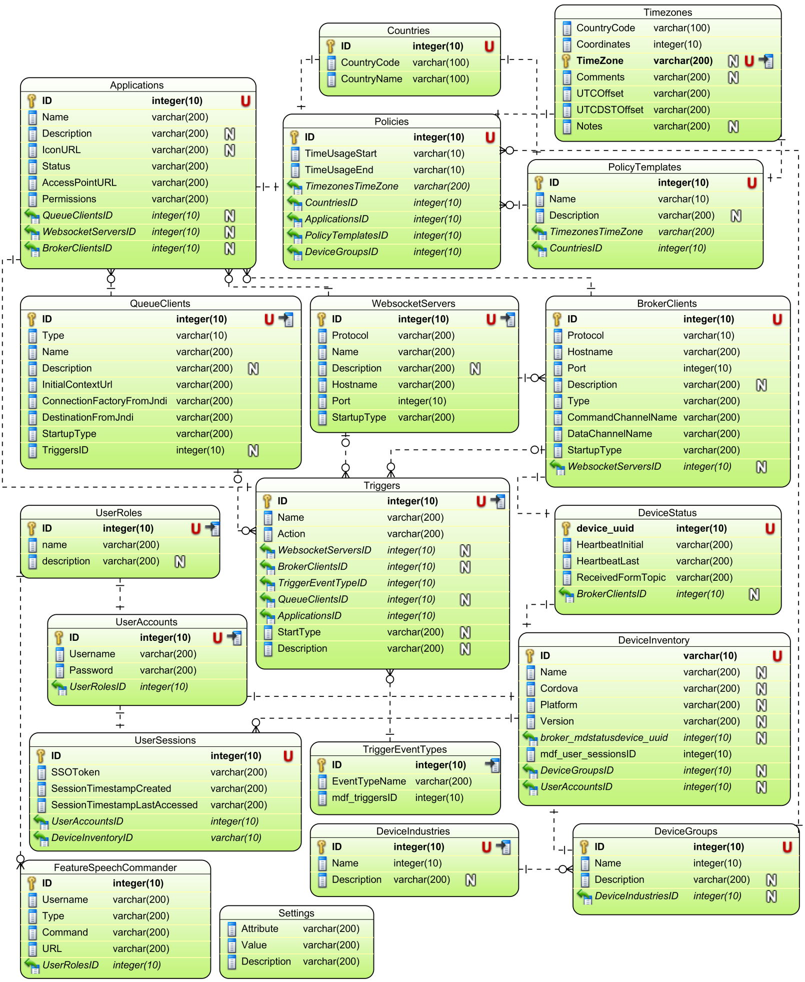


Figure 6.2: Relational database model

Table	Description
Applications	Register MDA applications and configuration
Policies	Associated rules set of the MDA application
PolicyTemplates	Template of a set of rules for a MDA application
Countries	List of countries and codes
Timezones	List of timezones
QueueClients	Configuration of queue clients
WebsocketServers	Configuration of WebSocket servers
BrokerClients	Configuration of MQTT broker clients
Triggers	Definition of MDF triggers
TriggerEventTypes	Definition of event types for MDF triggers
FeatureListeners	Feature listener configuration for continuous data streams of mobile features
FeatureTypes	Feature types of a feature listener
UserAccounts	Login data of the internal MDF user management (if no external LDAP service exists)
UserRoles	Definition of the MDF user roles
UserSessions	Internal managed user sessions of the MDF
DeviceStatus	Initial and last received heartbeat from mobile devices
DeviceGroups	Grouping of mobile devices by organizational unit
DeviceIndustries	Grouping of mobile devices by industry
DeviceInventory	Registered mobile devices
FeatureSpeech-Commander	Assignment table of the feature Speech Commander
Settings	General configuration of the framework

Table 6.2: Tables of the database model

Table `Applications` contains all registered MDA applications. An MDA application is assigned to a policy, which is reflected in Table `Policies`. A policy represents a set of rules in context of the framework. For example, the rule set includes rules about granted locations and times for mobile devices in order to access MDA applications. These rule sets are stored as template in Table `PolicyTemplates`. A policy can be derived only from a single policy template. The data communication between mobile devices and the framework consider the types QueueClients, WebsocketServers, and broker clients in the data model. Each MDA application can claim a separate instance of a data communication type. Table `QueueClients` contains all definitions for accessing queues, which are hosted by the QueueManager of a message queuing system. The Table `WebsocketServers` contains all definitions for the instantiation of WebSocket servers as well as access the WebSocket instances. The `BrokerClients` table contains all definitions to access all the instances of message broker clients. Each of these data communication types can implement triggers for occurring events. In this case, each trigger is assigned to an MDA application. The definitions of triggers are stored in the Table `Triggers`. A trigger is always derived from one of the trigger types that are stored in Table `TriggerEventTypes`. Trigger types are, for example, `DEVICE_HEARTBEART`, `DEVICE_IN_-TIMEZONE`, and `DEVICE_IN_COUNTRY` that trigger specified actions (trigger actions) when an event occurs. Because all mobile devices must be registered to the framework, they are recorded in Table `DeviceIn ventory`. A mobile device is always assigned to a user (Table `User Accounts`) and to a group (Table `DeviceGroups`), while the device group belongs to an industry (Table `DeviceIndustries`). At regular time intervals, a heartbeat of the mobile device is received by the framework via data communication type and all information is stored in Table `DeviceStatus`. Table `Settings` captures all framework-specific configuration parameters that are necessary for general operations. While accessing MDA applications, the user context and credentials are always checked. Table `UserSessions` stores the MDA sessions to prevent the user from entering the same login information each time when accessing these applications. In case that no external user directory (LDAP) exists, all user data are managed in Table `UserAccounts`, while the user roles are defined in Table `UserRole`. The feature SpeechCommander stores the voice commands and actions of users in the mapping Table `FeatureSpeechCommander`. The ORM between DDD and RDBMS is contrary to the object-oriented approach of DDD. All stakeholders have to speak a common language, but also ORM-related performance aspects play a major role. In recent years, NoSQL[3] database systems have been developed that takes place a non-relational approach and do not use defined table schemas. Thereby building blocks (e.g. entities, value objects, and aggregates) of the DDD modeling can be stored directly without any object-relational mapping. However, such database systems play a minor role in the PLM environment, because the majority of PDM systems are currently based on RDBMS. Further research is necessary to use non-relational database management systems such as Apache Cassandra or MongoDB for the Mobile Feature Framework. The second subpackage `edu.kit.mdf.infrastructure.persistence.wmq` implements components according to the concept of message queuing for the transmission of messages on queues. The message format is defined by the framework, but can be replaced by other formats. Mobile features must implement this data format which is used for communication. In industry, the XML format has been established for structuring data in a text based form, since platform independent data exchange can be implemented in almost every programming language. In particular, the advantages of platform independence are of elementary importance through the variety of mobile operating systems. The queues are provided by the middleware software *IBM Websphere MQ*. The configuration and administration of queues are performed with the *IBM WebSphere MQ Explorer*. Alternatively, other MOM products[4] such as *Apache ActiveMQ* can be used too. The framework stores incoming messages in queues every time they cannot be delivered immediately to the mobile device. The causes for this situation can be a missing connectivity to the mobile device, maintenance, and defects. A similar situation applies to mobile devices that want to transmit relevant data (e.g. measurement data), but cannot be consumed and processed immediately by MDA applications. Once the MDA application reaches operational readiness, the stored messages can be picked up from the queue and processed by the MDA application. The configurations for the access to the queues are stored in Table `QueueClients`.

[3] Not Only SQL (NoSQL) is a new database generation which provides various concepts for data storing such as key/value stores, document stores, and graph database

[4] Message Oriented Middleware (MOM) describes middleware software which is used for asynchronous or synchronous communication between systems.

6.5 User Directory Integration

Due to the large number of IT systems in companies, the count of administrative tasks for the user management have to be centralized and also reduced. For this purpose, LDAP services provide a simple and secure access to user data through open and flexible interfaces (LDAP is specified in RFC[5] 4510 and RFC 4511). PDM systems provide in the OOTB state an internal user administration, but usually PDM systems are coupled with an LDAP service. This eliminates a double administration of user accounts and saves the user from remembering additional login data. Based on these findings, the integration of LDAP was carried out in the framework without an internal user management. In addition, the PDM system ENOVIA V6 was coupled with the LDAP service to use a common user management. In this reference implementation, the OpenDS Directory Server provides the user management. The base distinguished name (base-dn) `dc=kit,dc=edu` was defined in the directory tree that specifies the starting point in order to search users along the underlying objects (see Figure 6.3).

6.6 Infrastructure Components

The infrastructure components include all components that are not part of the domain model core, but support this model with basic functionalities such as logging, exceptions handling, and security. On the one hand, established libraries have been used for some of these components. On the other hand, the logging framework *Apache log4j* is used for all framework components. This framework, which has been established over the years as an industry standard, allows the categorization of log messages according to level of importance and can control the output messages through various types of appender. The communication between framework components is supported by the API Specification JAX-RS[6], which is provided by the reference implementation of the *Jersey*

[5] Requests for Comments (RFC) is a collection of publications by the Internet Engineering Task Force (IETF) and others describing technical Internet standards

[6] Java API for RESTful Web Services (JAX-RS)

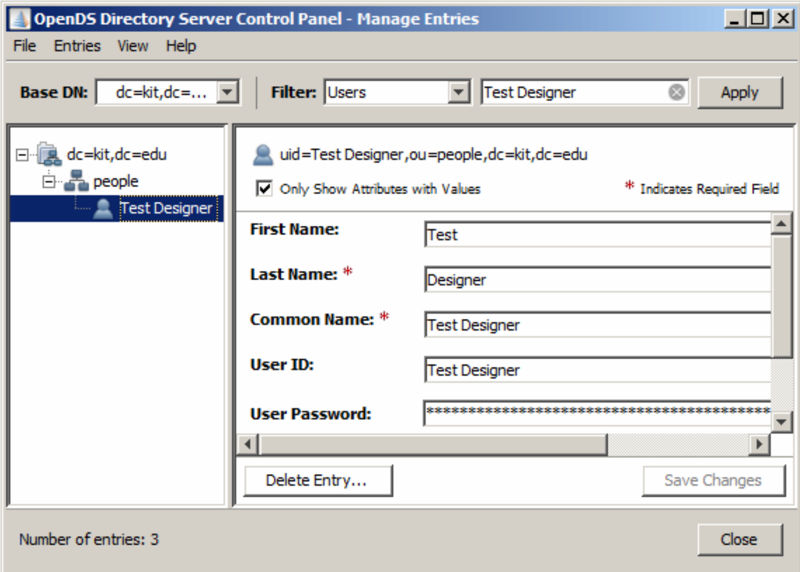


Figure 6.3: User management in the LDAP server OpenDS

project by *Sun*. The Jersey library supports the web-based communication over HTTP and can be integrated into an existing infrastructure of middleware components and into web-based applications such as MDA applications. JSON[7] is used as a text-based data format between framework components, since it has less overhead by more compact structure and simplified notation compared to XML. For this purpose, the implementation `org.json` by *json.org* is used. However, there are some other Java libraries such as *google-gson* and *Jettison*. The manufacturer of the database management system MySQL provides a connector for the data persistence. This library contains the data driver in order to establish a connection with the DBMS. Furthermore, a number of other libraries from the *Apache Commons* project are used to provide additional basic functionalities of Java through `commons-*` libraries. The best known are for example `commons-io` for Input-/Output-Streams and

[7] JavaScript Object Notation (JSON)

commons-lang for extending the standard Java functions java.lang. The WebSocket implementation of server and client has been directly considered in the framework. Therefore, the framework was built on the WebSocket implementation of *java-websocket.org*. This implementation is primarily used for mobile devices, which provide an appropriate mobile platform. This applies, for instance, to smartphones, phablets, and tablets. The MQTT implementation is based on the *Paho* project, which provides as open-source library (package: org.eclipse.paho) the *MQTT* messaging protocol in context of M2M communication and the *Internet of Things*. The MQTT protocol is especially important for mobile devices as a message protocol to transfer data when no extensive mobile platform such as *Google Android* is provided and also when the device is designed within its sphere of responsibility to fulfill specific tasks. The proprietary libraries com.ibm.mq.* are used to access the message queues of the MOM software *Websphere MQ* by IBM to temporarily store messages on the queues. The MDC implements few core components of the mobile development framework *PhoneGap*, which were heavily modified and extended for the Mobile Feature Framework model. PhoneGap focuses on the development of hybrid mobile applications, in which the user interfaces and logic are implemented with *HTML5*, *CSS3*, and *JavaScript*. The *FFI*[8] is applied to interact with natively implemented and platform-specific features using the JavaScript programming language. FFI allows function calls that have been implemented in other programming languages. The MDC does not use FFI for feature interaction compared to PhoneGap, but preferable the protocols MQTT and WebSockets. These protocols are necessary for the Mobile Feature Framework in context of PLM, because the server-side implementation and deployment takes place exclusively by Web-based MDA applications and thus supports the component based concept of PLM as well as harmonizes in conjunction with direct integrations in the PDM system. In terms of the fourth industrial revolution, it is necessary to use these new communication protocols for the feature integration to ensure the seamless communication capability of mobile devices between networks. This also includes the infrastructure of devices based on cyber-physical system, so that the presented concept

[8] Foreign Function Interface (FFI)

and framework in this research work is also suitable in context of cyber-physical system. In addition, the MDC and implemented features have been specifically designed and developed for the PLM context with focus on industrial use, while PhoneGap favores a general orientation. The PLM-specific focus applies to features such as object identification by image recognition, PLM object-based layers for map services (e.g. localization of products, assemblies, machine, and project staff), and voice commands to call PLM functionalities. This framework also supports the concept of predictive apps to act with foresight with the user using MDA applications. Further advantages of the Mobile Feature Framework concept are the server-side administration and control of user interactions. The MDA and MDM implement for user interface the web framework *jQuery Mobile*, which was developed especially for touch-optimized screens. The main advantages of this framework are the browser-compatibility with all major mobile platforms, optimization for browser rendering engines, and scalability for different screen sizes of mobile devices. For this purpose, the *App Framework* and *Sencha Touch* are an alternative implementation, which provides an improved performance in certain scenarios. The PDM-related MDI component uses proprietary API libraries of the manufacturer in order to interact with the PDM system. For this purpose, the PDM system *ENOVIA V6* by *Dassault Systèmes* was used as repository for product data, which require the packages `com.dassault_systemes.platform.rest.Services` for RESTful web services, `matrix.db.MQLCommand` for querying business objects and relationships using MQL as query language, `matrix.util.*` for calling diverse basic functions of the PDM system as well as other packages. MQL[9] commands and API calls are used to retrieve crosslinked business data within a valid user context.

9 Matrix Query Language (MQL) is a SQL related query language used in ENOVIA

6.7 Framework Testing

The unit testing framework *JUnit*[10] was used to reduce the susceptibility to errors of Mobile Feature Framework components. JUnit is especially suited for automated test procedure to verify classes or individual methods (so-called units) by defined test cases. An individual test case, which validates the functionality of the each component, has been developed for each unit to be tested. The result of the test cases can only engage the states, *passed* or *failed*. Figure 6.4 shows an example of the validation for the object `BusinessObjectType`. Therefore, the separate class `BusinessObjectTypeTest` was generated to implement the individual test cases. Figure 6.4 illustrates eleven test cases that have been implemented only for this single unit and have been successfully completed with a pass status. The green status indicates that the behavior of the tested component is unchanged and correctly implemented, while a red status would indicate an unintended change in behavior of the component. The implementation of the MDM can also be considered as an additional unit testing component, because the framework interface is used for administration of MDA applications, broker clients, WebSockets, and mobile devices. Thus, the cross-component tests have been already carried out and validated for the framework by various MDM scenarios (e.g. registration of MDA applications, configuration of broker clients, send and receive feature messages). In addition, a variety of MDA applications has been implemented by invoking various mobile features of the mobile device and validating the returned results. Figure 6.5a shows the user interface of the implemented MDA applications on a smartphone and Figure 6.5b shows the web-based administration interface of the MDM on a stationary workstation.

[10] JUnit is an implementation for automated software tests for the Java programming language. xUnit is used as a generic term for implementations of unit testing frameworks in various programming languages.

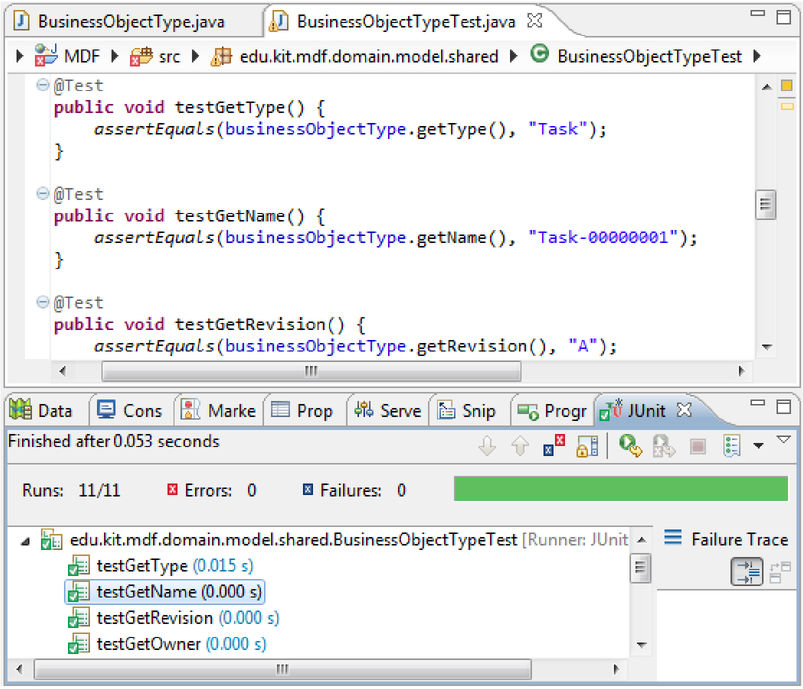
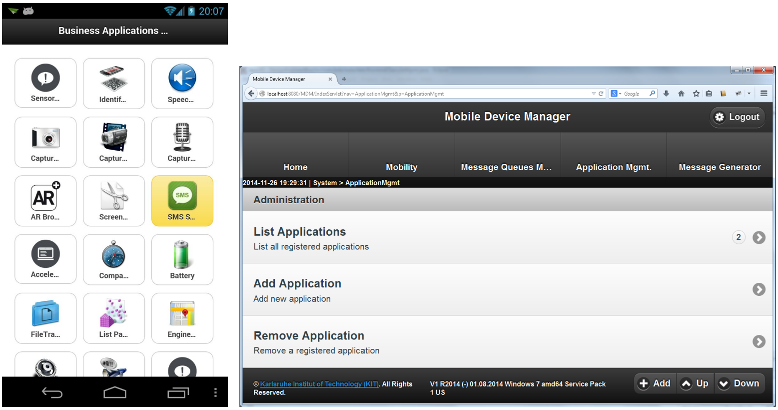


Figure 6.4: Validation of framework components using JUnit

6.8 Summary

The main objective of this chapter was to represent various aspects of the framework engineering in context of the Mobile Feature Framework. For this purpose, the development environment was presented in Section 6.1 and the project structure has been designed and consistently implemented in Section 6.2. The framework structure with predefined package and subpackage for several applications domains was presented and classified in Section 6.3. Section 6.4 covered the persistence of data objects in the framework, which was achieved by a relational database model. Based on this database model, the respective entities and relations of the framework have been described. Subsequently, the integration of external user directories was discussed in Section 6.5 in order to achieve a centralized user administration. Moreover, further infrastructure components were used for the framework and explained in Section 6.6. A test framework has been presented in Section 6.7 to validate and minimize the fault susceptibility of framework components.



a) *MDA* **b)** *MDM*

Figure 6.5: User interface of MDA and MDM

The approach, which has been adopted for implementation of the framework represents only an option of a plurality of ways. The conceived development environment and project structure were largely determined by the previously designed domain model. However, the persistence of data objects is independent from the domain model, so that other approaches for persistence can be selected as well. This also applies for the used infrastructure components such as logging and user directories. The validation of framework components can also be performed with alternative testing frameworks. The selection of the testing framework was dependent especially on the used Java programming language of the realized framework components. In summary, it can be said that a wide range of existing technologies are available for the framework engineering which significantly reduced the effort and cost of implementation.

However, a well-founded basis must take place for decision making in technology selection for industrial contexts to avoid discrepancies and risks in an enterprise specific environment. The basis for decisions is commonly influenced by existing technologies and infrastructures as well as various company-specific assumptions. It is also essential to ensure long-term investments. Therefore, each company has to cope with other individual challenges of technological application. The present implementation represents a reference for the integration of mobile features in practice, since the practical feasibility of companies depends on a variety of factors such as total number of mobile users and feature requests, amount of transmitted mobile data as well as performance of systems and networks. These aspects are considered from an operational and architectural point of view by infrastructure teams.

7 Framework Application and Valuation

This chapter deals with the application and evaluation of the Mobile Feature Framework in context of PLM. For this purpose, two different application scenarios are described. Each use case is designed in a separate MDA project and implemented as a prototype through a mobile MDA application.

7.1 Objectives

The evaluation of the framework concept takes place with respect to the fundamental feasibility of the mobile feature integration in mobile applications. In addition, the overall approach of the mobile feature integration for the PLM context is generally evaluated. For this purpose, the primary question that this study will raise is, to what extent the architecture approach of the framework supports the integration of mobile features to meet the requirements of Chapter 2. Furthermore, future challenges are identified by mobile features applications that arise in a widespread industrial application and the challenges to be met are explained through appropriate solutions. On the basis of this discussion, the problem fields are identified and measures are derived for supporting the further development and acceptance of the overall framework.

7.2 Selection of the Evaluation Methodology

The optimal method for the validation of the framework is the implementation of a customer-focused mobile application that uses mobile features in context of a complex PLM landscape. However, this validation method simultaneously imposes a number of extensive requirements as well as project-specific dependencies that exceed the effort in context of this research work. These requirements refer to the following dimensions:

- **Project duration:** PLM projects require in some cases several years due to the high number of integral business processes. In addition, the perspectives of many different users and the integration of specific IT systems must be considered. However, the full implementation is only one aspect, because the introduction of the application also requires an adequate training and an acclimatization period for the user. Therefore, the benefits can be measured and examined only after a prolonged time.
- **Effort and cost:** This research focuses on the generic integration of mobile features for PLM. In contrast, PLM projects are always carried out in a specific customer context, since companies of the same industry define and implement rarely identical business processes. However, the consideration of customer's context is necessary for a practical test application in an industrial-oriented PLM project. For this purpose, the identification of business processes must be carried out. Each company has established its own processes and procedures over recent years. Moreover, it is necessary for the conception of the application to involve the respective experts who have the knowledge about industry and company-specific business processes and work from various business units of the company with the consequence of additional expense for costs and resources. In addition, high standards of safety, reliability, quality, and adaptability from the existing IT infrastructure of the customer are placed to the framework that exceeds the effort of this research work for the validation of the framework. Therefore, the realization of a practical application in the context of a customer is not affordable and thus not further pursued in this work.

- **Introduction:** It is necessary to identify the concerns and prejudices of users prior in a early stage of the design phase of the application. This can be implemented before designing and implementing a new test application. Users must be determined by a deliberate survey so that their individual privacy concerns can be responded. Subsequently, the results of survey are used to address the requirements to the customer-specific test application in the design phase and thus to resolve privacy concerns of users. Thereby, the acceptance of the test application that uses mobile features can be increased. It is necessary to offer an appropriate training for the users while introducing the test application. For this purpose, training material and practical exercises must be created and verified.
- **Risks and habits:** Applications that use mobile features require a change in approach and mindset of users. Therefore, the resistance and rejection of users regarding mobiles features must be expected, especially when extensive user-related data are collected. The combination of information and mobile features simultaneously produce several new application scenarios that seem unfamiliar, outlandish, and have to be learned intuitively by the user. The risk of falling back into old patterns of behavior and continuing already established habits must be reduced to a minimum. A typical behavior pattern is, for example, to print out information rather than querying information using a mobile device. In this case, the results of a sociological survey could support the design phase of an application.

The introduction of a practical and comprehensive test application, which considers the individual customer requirements cannot be covered in context of this research work and is not further intended. Instead, test applications are implemented based on fictitious application scenarios related to industrial needs of the PLM industry. Fictitious application scenarios are conventional and frequently encounter situations that have a practical relation to PLM processes. Such fictitious situations are described for the validation of the framework by the PLM applications ShipRepairAssistant in Section 7.3.1 and MyDesignerFellowshipTools in Section 7.3.2.

7.3 Case Studies

Validation of the framework requires the description of two different fictitious situations which have a practical link to industrial PLM processes. The respective situation with the problem description is formulated, the concept with the objective is defined, and subsequently the implementation of mobile test application is carried out. For this purpose, the test application implements mobile features of the framework. Finally, the evaluation of mobile application takes place to ensure an adequate practical use. Furthermore, the resulting advantages, technical limitations, and potential social impacts are discussed for mobile users. Use Case 1 deals with the optimization of the operation and maintenance sequences of shipbuilders. High requirements of the owners put shipbuilders in the early stage of the design under tremendous pressure to reduce the cost of operation and maintenance to a minimum. However, all components are continuous subject to wear by chemical and thermal stresses. The consequence of the stress over time is related to the interference and potential failure. Thus, the substitution of components cannot be avoided, but the replacement process of spare parts can be made more efficiently. High safety and environmental requirements assist the technicians in the risk-minimized execution of tasks in the ship. Use Case 2 deals with the efficient communication capability of design engineers in mobile situations for the machinery and plant engineering sector. In view of the intensified competitive situation, manufacturers of this industry must provide smarter and more efficient plants and equipment for customers in order to maintain their competitiveness against competitors. However, this will only succeed if design engineers have a high level of creativity and develop new concepts by sharing and exchanging ideas with other people. In order to support Design Engineers in this early phase, the ability to communicate must be present also in mobile situations. Mobile features support Design Engineers therein to make contributions and consume mediatized information which is tailored to the individual mobile work environment.

7.3.1 Business Case: Assistant for Customer Technicians

In the first application scenario, the maintenance service department of company Bergmann Ship Service GmbH, specialized in ship repairs, receives an order regarding a faulty valve in the engine room. The customer request does not include any additional information about defect components which may need to be replaced. Firstly, the defective valve must be located and identified by the customer technician on site. Secondly, the technician must verify, whether the defective valve is repairable or has to be replaced. In case of exchange, a suitable replacement part must be identified and the nearest logistics center with the available spare part has to be found. Thirdly, the spare part order is triggered and a further appointment with the customer is arranged in order to replace the defect part. Once the part is available on site, the exchange process is performed at the confirmed customer appointment considering the applicable safety regulations. The use case can also be considered from the SLM perspective; however, this work primarily focuses on PLM.

Problem Statement

Most of the maintenance services have appropriate tools for system monitoring and problem analysis in order to detect errors in assemblies. However, some upstream tasks of the problem analysis such as locating defective components are insufficiently covered by these tools, although the access to the IT system (e.g. PDM system) would be possible. Therefore, the technician has to request for guidance to the local staff, who accompanies him to the component. Subsequently, the technician must identify the corresponding component in order to perform the error analysis using the maintenance manual. Problems during the identification of the component often occur when the label is attached to surfaces that are inaccessible for the technician, the label has been damaged, or is no longer present. This causes a delay in the workflow, because the problem analysis cannot be started without prior identification of the component. In this case, additional information must be obtained from the local staff. The identification of components by image recognition and ID codes would avoid such delay in the workflow. However, the current problem analysis tools do not have appropriate identification functionalities such as image recognition. Moreover, no additional information about the actual context of the component is determined when it represents only a single component of a complex system. This has the consequence that a perceivable and proactive support of the technician takes place insufficiently during the repair process in consideration of applicable safety regulations.

Conception

The objective of the ShipRepairAssistants is to assist technicians in the execution of the workflow by the use of mobile features. Through the combination of mobile features with context information of the component, workflows can be represented more transparently and simplify as well as reduce delays to a minimum. Thus, the ShipRepairAssistant can be considered as a complement to conventional problem analysis tools to cope with generic tasks (e.g. localization and identification of parts), while problem analysis tools focus on specialized tasks of root cause analysis. The technician obtains presented relevant information of the component in the actual valid mobile context, regardless of the use of stationary workstations. The workflow is thereby more efficient for the technician and an optimization of the operation and maintenance sequences is reached. In order to achieve these objectives, it is essential firstly to determine, which of the mobile features offer technicians true added value in the individual steps of the work sequence. Table 7.1 illustrates the mobile features used for the ShipRepairAssistant with the resulting benefits. The concept of the ShipRepairAssistant considers various subareas of the technician as illustrated in Figure 7.1. The subareas differ according to direct and indirect interactions with objects (e.g. part) performed by the technician. With reference to the defined subareas, a component can be localized, examined, related information can be presented or contributed and object-related actions can be executed. Each group uses respectively other mobile features, which have already been listed in Table 7.1.

Generated Added Value	Used Mobile Feature / Use Case
Localization of the Assembly	
Navigation to the part	Feature modeled in Section 5.3.3 using Map service for object localization by coordinates
Spatial orientation of the part	Feature modeled in Section 5.3.3 using augmented reality browser
Identification of Parts	
Part identification by barcode	Feature modeled in Section 5.3.1 using barcode scanner library
Component identification by image recognition	Feature modeled in Section 5.3.2 using recognition service
Execution of the Repair	
Calling und navigating through the guided fault analysis by speech	Feature SpeechCommander modeled in Section 5.3.4 using voice recognition and accelerometer
Simplified error analysis of the faulty assembly by noises	Feature described as pattern in Section 4.4.2 of Group *Identify Objects*
Procurement of Spare Parts	
Spare part localization	Feature modeled in Section 5.3.3 using map service for object localization by coordinates
Repair report creation	Feature SpeechCommander modeled in Section 5.3.4 using voice recognition and accelerometer

Table 7.1: Used mobile features in application ShipRepairAssistant

Implementation

The previously developed SRA concept is now applied in the implementation phase. Three main categories from the design description are derived for the mobile application implementation.

- Part Management: Identification, recognition, and replacement of components and machines

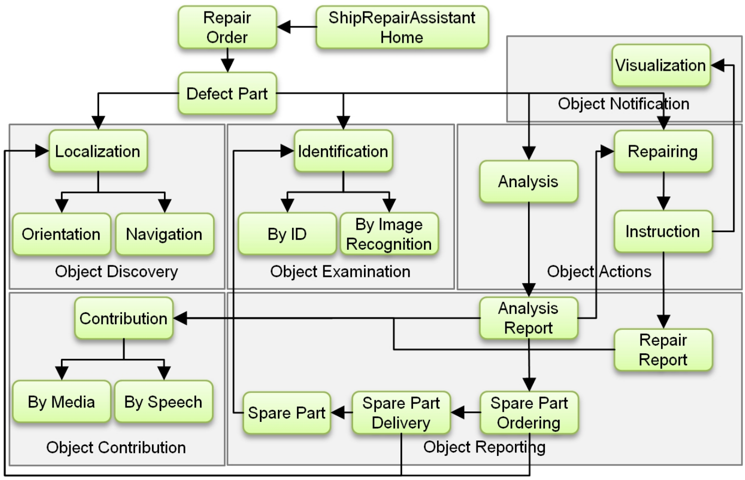


Figure 7.1: Concept for application ShipRepairAssistant (SRA)

- Service Planning: Localization of field engineers, defective components, and spare parts
- Reporting: Creation of repair reports

All areas are not implemented to work isolated from each other but are integrated into each other under the aspect of service management. For example, in a sequence of steps it is necessary to perform the localization of defective components as first activity before carry out the component identification. Figure 7.2 shows integrated mobile features of the Mobile Feature Framework for localization and identification of components. The necessary PLM data of the components are requested over the MDI and obtained from the PDM system to provide the mobile features with the required component information. The data preparation for mobile features such as part map and AR environment of technicians take place by a layered JSON structure in which a layer is represented by exactly one object type. The detailed information enables the adaptation of the mobile features (Map/AR) for the technician context. Subsequently, the fault diagnosis can be carried out to initiate the replacement operation of the defective component based on the fault results. Finally, the

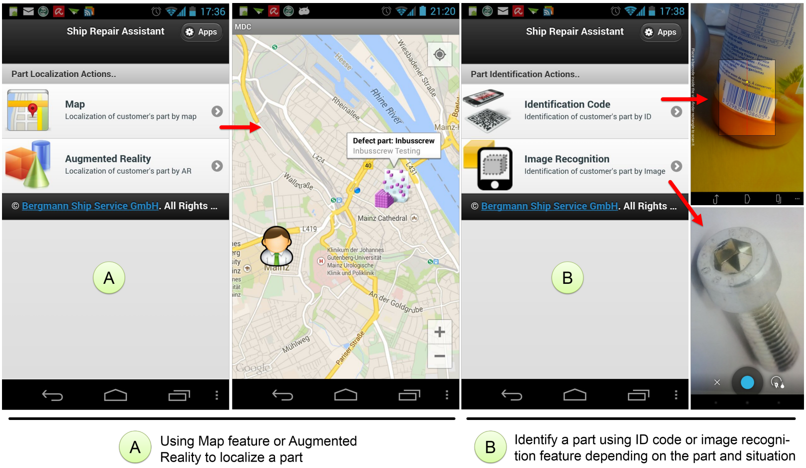


Figure 7.2: Localization of parts in SRA application

creation of a repair report supported by media mobile feature takes place to capture the results. The functions of the SRA application can be accessed by the technician either by conventional page navigation or voice-based navigation. The featurization of SRA webpages must be performed in order to integrate mobile features of the framework for the technician context. A relational database schema has been created for the SRA application, which stores the repair orders, repair reports, and service requests of the customer. In industrial environment contexts, this kind of information is normally retrieved from the ERP system, CRM system, and SCM system such as PeopleSoft or SAP. However, the implementation of interfaces to these systems is not carried out for this use case, since the focus is set on the validation and valuation of mobile features in PLM context. Figure 7.3 shows the information capturing for the repair report through a featurized webpage of the SRA application. The data capturing takes place in a language-based through the technician. Therefore, the technician taps the appropriate form field in order to invoke the voice recognition of the mobile device. Once the technician has spoken a word or sentence, the proposed text of the speech recognition is transmitted from the MDC to the MDF which forwards the results to the SRA application. The SRA webpage recognizes the addressed form field based on the featurization process. The mobile feature request remembers the user context in which the feature call was performed. This mechanism provides the possibility to perform appropriate interaction with the form that is currently displayed to the mobile user. In addition, media features such as audio recording, image capturing, and video recording are used to capture detailed facts in reports for better understanding. The technician can save a lot of time filling out a repair report with feature-based information capturing compared to keyboard-based capturing using input fields. Thus, the technician should have a detailed documentation of customer orders and can perform operations more efficiently. However, an empirical investigation in industrial environments with service technician must be conducted to confirm the conclusions of the author.

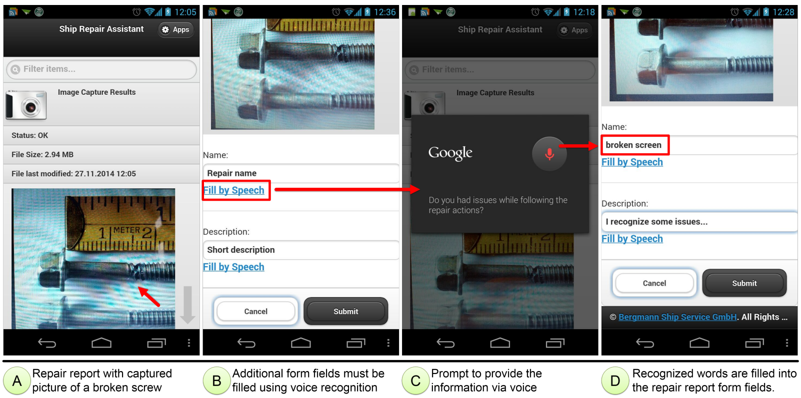


Figure 7.3: Voice-based information capturing in the repair report

Results and Benefits

The mobile application ShipRepairAssistant has demonstrated that the use of mobile features for various industrial contexts can provide a specific added value. In this particular scenario, the application is an ideal supplement in order to provide context information for the technician who would otherwise have to collect information manually from various sources or would be plainly missing during the working steps. Thus, the technician has a flexible tool at hand that helps to close the information gap and to cope with working steps more efficiently.

7.3.2 Business Case: Activity Guide for Design Engineers

In this application scenario, design engineers of the engineering company Hans Meyer GmbH must continuously develop creative ideas and elaborate within the team. The pressure for innovation is correspondingly high. The company is in tough competition with other competitors in the industry and is therefore increasingly dependent on innovative ideas of Design Engineers. In order to increase the creativity of design engineers, the company allows employees flexible working hours and location. Therefore, employees are not necessarily dependent upon the office space of the company and can thus favor the development of creative ideas by external influences. The design engineer uses as a source of his inspiration various locations outside of the office space on company premises as well as various external locations during free time. The design engineer obtains a mobile application installed on his existing smartphone. This application supports him not only in capturing and sharing of ideas and communication to members of the design team but also in daily life activities due to the stationary absence.

Problem Statement

Nowadays, design engineers use for the exchange of information mostly stationary workstation. The communication via mobile devices takes place relatively rarely and, if at all, then only predominantly by e-mail. However, the need for information availability in specific locations exists but which can only be satisfied insufficiently. In addition, innovative design ideas are not always develop at the stationary workstation, but rather at different inspiring places and commonly in dialogue with people. Any additional information that would help to develop new spontaneous ideas cannot be immediately retrieved and communicated to other people.

In addition, spontaneous ideas and thoughts are quickly forgotten unless they are written down. In order to close this information gap and also to change the communication focus in favor of mobile devices, design engineer is given the application to support him in mobile situations.

Conception

The objective of DesignEngineerBox is to support the design engineer in the creativity phase by the use of mobile features which help the design engineer to capture ideas or any kind of information using various media types and new technologies as well as share, discuss, and modify content with team members. In case that the digital communication with persons of the team is insufficient to discuss a subject, a map of current locations of the team members is visually presented to the design engineer. This allows him to organize unbureaucratically meetings with team of people who are actually available at company's campus and subsequently to capture the results in the PDM system straightforward from the mobile context. In addition, upcoming activities and actions of the Design Engineers are displayed on the smartphone and can be supplemented by media contributions (e.g. audio, video, images), regardless of the use of the stationary workstations. Thus, the daily routine of design engineers is more efficient and also promotes the creativity of the person. In order to achieve these objectives, it is essential firstly to determine, which of the mobile features offer design engineers a true added value in the creativity phase as well as in workflows to cope with everyday tasks. Table 7.2 illustrates the mobile features used for the DesignEngineerBox with the resulting benefits. The concept of the DesignEngineerBox considers various subareas of design engineers as illustrated in Figure 7.4. The subareas differ according to performed interactions with objects by design engineers. With reference to the defined subareas, the objects "Idea" and "Issue" are either captured, examined, related information presented, or contributed. In contrast, the object "Person" is localized and visualized. Each group use respectively other mobile features, which have already been listed in Table 7.2.

Generated Added Value	Used Mobile Feature / Use Case
Localization	
Localization of team members	Feature modeled in Section 5.3.3 using map service and augmented reality
Navigation to employees and rooms	Feature 5.3.3 using navigation service
Spatial orientation in the building	Feature modeled in Section 5.3.3 using augmented reality browser
Highlight employees when they enter company premises to save time	Feature modeled in Section 5.3.3 using map service, vibration and speaker for people notifier
Idea Management	
Capture and edit images	Feature described as pattern in Section 4.4.7 of Group *Contribute Object Content*
Image sharing with other team members	No mobile feature required (Application's responsibility)
Evaluation of ideas by human senses (e.g. speech)	Feature described as pattern in Section 4.4.1 for gesture, speech, and voice recognition as well as modeled as feature SpeechCommander (see Section 5.3.4) using voice recognition.
Consideration of the user context by day-night mode	Feature described as pattern in Section 4.4.1 of Group *Observe Objects* using the light sensor to detect the light intensity
Activity Management	
Calling everyday tasks by speech	Feature SpeechCommander modeled in Section 5.3.4 using voice recognition and accelerometer
Control objects activities by speech	Feature SpeechCommander modeled in Section 5.3.4 using voice recognition and accelerometer
Assist object creation (fill form-fields by speech)	Feature SpeechCommander modeled in Section 5.3.4 using voice recognition

Table 7.2: Used mobile features in application DesignEngineerBox

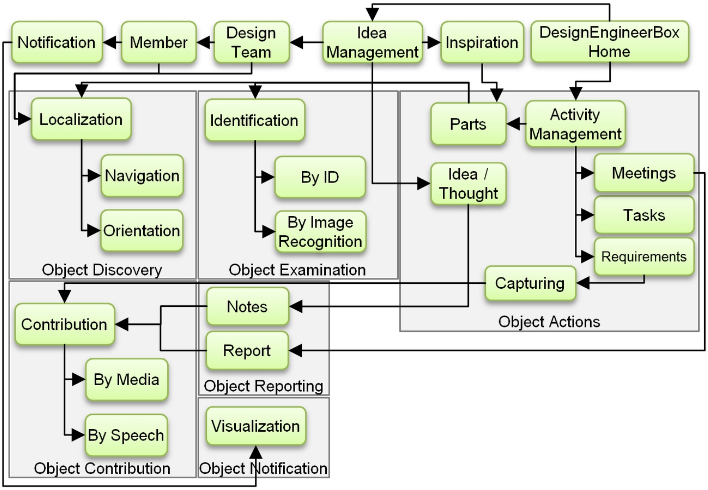


Figure 7.4: Concept for application DesignEngineerBox (DEB)

Implementation

The previously developed DEB concept is now applied in the implementation phase. Three main categories from the design description are derived for the mobile application implementation.

- Activities: Publicly visible activities and tasks of the mobile PDM user
- Teams: Collaborative communication and interaction within teams
- Idea Scratchpad: Capturing and sharing of private knowledge and ideas

All parts of the application are not implemented to work in isolation from each other but are integrated into each other. For example, captured ideas that were initially accessible only for the design engineer can be shared and discussed with other people in the team. The individual functions of the application can be called either by conventional webpage navigation or voice-based navigation. The voice-based navigation and control require the featurization of DEB webpages to integrate the mobile feature of the Mobile Feature Framework in the design engineer context. In addition, a relational database schema that reflects the relationships of private ideas, attachments, and teams have been created for the DEB application. The people have been identified over the MDI from the PDM system in order to create a team and to retrieve person object-related information (e.g. tasks, issues, and meetings of the mobile user). The retrieved data from the PDM system are used to supply the mobile feature with required information. The detailed information supports the adaptation of mobile features to the current Design Engineer context. Figure 7.5 shows featurized options for a person. Therefore, the feature data (e.g. for Maps/AR feature) are prepared in the DEB application and subsequently a mobile feature request is created and sent to the MDF.

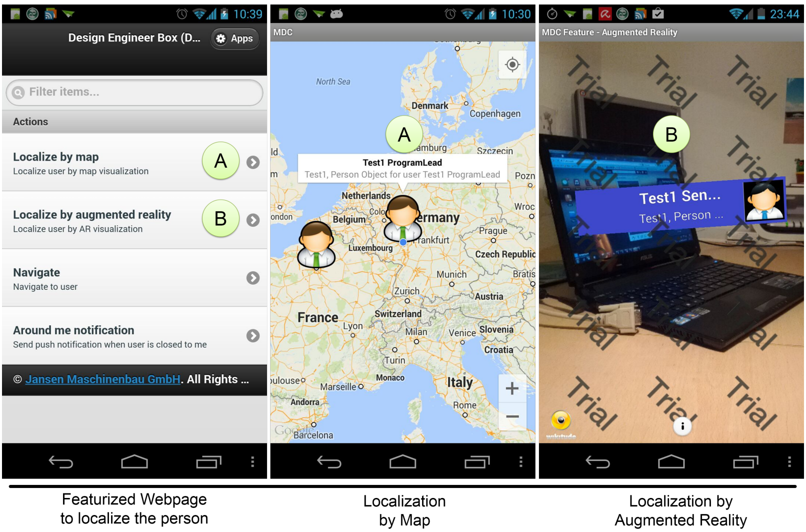


Figure 7.5: Featured actions for person objects in DEB application

The data preparation for the map and AR feature takes place through a layered JSON structure in which a layer is represented by a single object type. For example, the first layer contains all persons and the second layer contains all locations of captured idea objects. Figure 7.6 shows featurized choices for ideas and issue capturing, e.g. language-based lifecycle change (Promote / Demote) of Issue objects. The user calls on the Issue object and shakes the mobile device. The shaking event triggers the MDC to send a request to the MDF which is forwarded to the DEB applications. The DEB application sends a feature request over the MDF to the MDC to enable a feature for the current design engineer context.

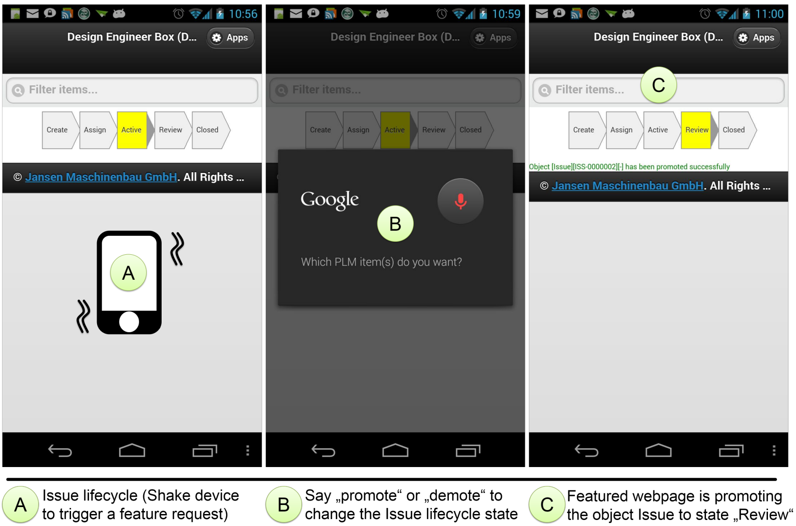


Figure 7.6: Capturing and control of objects in DEB application

Thereby it is possible to perform appropriate actions on the Issue object that is currently displayed to the mobile user. Thus, the displayed Issue object ISS-000002 of the DEB application can be promoted to the next lifecycle state "Review" by shaking once the mobile device followed by the voice command "promote". The object will be demoted to the previous lifecycle state while shaking once the device followed by the voice command "demote". Additionally, text-based form fields of new Issue objects and ideas can be filled using the speech recognition feature. Thus, the design engineer saves a lot of time and can prevent typing errors when they use these features for text-based input fields and thus should increase the acceptance of the information contribution in mobile situations. The higher acceptance by users through mobile features still needs to be examined more closely in future studies with test groups.

Results and Benefits

The DesignEngineerBox has demonstrated that the use of mobile features supports design engineers in the communication of innovative ideas compared to conventional stationary tools. This scenario is not limited to the industrial context of design engineers and can be transferred with minimum effort to various industry branches. Thus, the application is an ideal complement to applications on the stationary workstation in order to share relevant information within the design team in mobile situations. Otherwise, design engineers would have to communicate the captured information by conventional stationary work tools (e.g. e-mail, messenger, phone), which limit and potentially reduce the innovation of design engineers. Thus, the design engineers have a flexible tool at hand that helps to overcome the social, technical, and process-intensive barriers through mobile features, which lead to a higher innovativeness of employees.

7.3.3 Valuation of the Business Cases

The presented fictitious applications cover completely different subject areas in the PLM environment. While the first use case has focused on the optimization of operation and maintenance sequences of shipbuilders, the second use case has addressed the innovation ability of design engineers. However, both scenarios represent only a part of the Mobile Feature Framework capabilities. The possible diversity of device types, hardware features, mobile features, and scenarios allow accordingly an enormous number of possible combinations. The application examples have just demonstrated a small selection of scenarios that can be implemented using the framework. Not all mobile features that have been mentioned in the application examples have been fully implemented. This is justified by the complexity of the underlying infrastructure for some features. For example, mobile features for the recognition of objects by sounds and images require the creation of different databases with samples, the modeling of appropriate logic structures as well as the definition of feature interactions. Accordingly, all features must be developed conceptually in which the Mobile Feature Framework provides the frame for implementing an extensive range of features for the respective scenarios. These fictitious applications were chosen because the author has had many years of practical experience in these topics gained through several customer projects. The tasks for servicing and predictive maintenance of equipment were rudimentary implemented for technician service visits in the banking and insurance sectors. The experience of the author in technical support operations, however, showed that an optimization of work processes had been able to increase the efficiency of employees clearly in performing tasks by mobile features.

7.4 Summary

The application of the Mobile Feature Framework in context of PLM has demonstrated the scientific benefits through two different applications, ShipRepairAssistant and DesignEngineerBox. These were preceded by the explanation of the objective and the selection of a suitable methodology in order to perform the validation. In this regard, two different use cases were selected and the problem statement, conception, and implementation were described for each case separately. Subsequently, the requirements of the test applications were derived from the situational contexts of the social and business perspective. The use cases have shown that the provision of mobile features by the framework appears to be appropriate in order to deploy mobile features flexibly, standardly, and uniformly for various scenarios. The use of the Mobile Feature Framework allows the simplified and more efficient creation of mobile applications using mobile features without redefining and implementing feature-specific components from scratch. The provision of mobile application functionalities with mobile features achieved a significant improvement in workflows and collective collaborations in distributed teams for mobile users. In addition, the provision of information in mobile situations was significantly increased and hence the proper task execution was supported proactively. The evaluation of the Mobile Feature Framework takes place based on the defined requirements in Chapter 2. The requirements differ from the social, business, and informational perspective. While social aspects addresses the privacy, user context, and cultural needs, the business aspects focus on the respective views of user roles for the fulfillment of tasks. The informational aspects consider the technological requirements for systems and networks. The component-based framework has been equipped with basic functionalities and has provided the mobile features and mobile applications with a high degree of stability. A wide variety of mobile features for the integration in mobile applications is provided to developers without the need of additional integration efforts on mobile devices. The efficient provision of mobile features is ensured through reuse of existing feature patterns from the framework. Moreover, feature requests are efficiently processed by the framework and routed to the mobile device. The communication protocol used for the data transfer between the mobile device and framework can be selected for each mobile application individually. The transmission time depends largely on the data volume, whether a mobile feature requires larger amounts of data, or whether larger amounts of data need to be transmitted from the mobile device to the framework. The efficiency of (mobile) networks used for the communication is also a crucial aspect for frictionless interactions with users and cannot be influenced by the framework. However, network interruptions can be recognized and appropriate actions are initiated by the framework.

The reliability of server-based framework components was already verified in Section 6.7. In so doing, the robustness and fault-tolerance of individual components have been validated by appropriate tests which were proven with positive results. The security of the communication between framework components is ensured by secure communication protocols. In addition, a security model for the framework was presented and implemented in Section 4.9.2. The extensibility and modifiability of framework components and mobile features are supported by the layer-based and component-based architecture. Infrastructure components can be replaced modularly through the separation of the business logic from the infrastructure in layers. The modularization has been considered for mobile features and also covered by the *Device Feature Extensibility Model* in Section 5.2. New mobile features can be developed and registered to the framework as a self-contained module in form of an additional plugin. Likewise, existing mobile features can be extended and modified as well. The scalability of the framework is assured by the separation of communication channels to mobile devices. In addition, the access address to be used for the mobile application is already provided to the framework during the registration process of the application. The framework uses this access address to control the data flow between the mobile device feature and the mobile application.

The maintainability of the framework depends greatly on the documentation of the interface specification. Likewise, the component-based design is an important aspect in order to implement future changes in the framework successfully based on self-contained functional modules. The portability of the framework to other mobile device classes is given but requires the implementation of basic functionalities through an access client for each mobile platform individually. In case that the mobile device does not provide a full-featured mobile platform, rudimentary functionalities can be implemented to allow communications and basic interactions. However, this implies that the mobile device requires a network connection and acts intelligently. In addition, libraries and protocols were used for the framework, which is either standardized or has represented a de facto standard in the industrial context. The standardization also significantly supports the interoperability to allow a uniform communication between mobile devices and the framework. The openness of the framework becomes apparent for the provision of PLM data by the integration module. In this case, any integration module can be developed and coupled with the framework (e.g. ERP system and MES system) to ensure the flexible provision of information for mobile features and mobile applications from other software systems. The framework components and mobile features have been continuously developed and extended during the realization of the prototype. Due to rapid technological changes, it is necessary to extend these activities in the future to continuously expand the range of mobile features provided by new mobile device classes.

8 Summary and Outlook

8.1 Summary

For many years, PLM-related tasks have been performed exclusively on stationary workstations. By introducing new classes of mobile devices, the mobile revolution began ultimately, which led that long-established patterns of thought and behavior were broken and existing processes have been accordingly revised or completely redefined. Existing processes have been accordingly revised or completely redefined. This meant that mobile situations could be considered now and will not be excluded from the outset anymore. The mobile revolution has initiated a change that defines new standards in the communication and interaction between people in various industrial fields.

Mobile features are insufficiently used for industrial purposes. However, the market demands that companies of various industrial sectors increase steadily the innovativeness and the efficiency of processes and operations so that products in a wide variety can be pushed in the market in ever shorter development cycles. Companies that do not bow to the market conditions lose competitiveness and will quickly fall behind. In order to prevent this case, free space for creativity and innovation must be given to employees. Since most creative ideas and inspirations arise spontaneously and are first discussed by the employees in the collective, mobile features support the employee in the media communication and interaction in mobile situations. In addition, data are collected mainly through manual input forms that have not been optimized for mobile applications. Furthermore, user-specific contexts are neither perceived nor taken into account in the application. Mobile features assist users in this situation in the data acquisition and consummation over all stages of the product lifecycle through context-sensitive information and innovative data interactions. The specific problem lies in a missing generic possibility to integrate features for PLM. The fact that the majority of mobile PLM applications have been implemented proprietary and natively, a unified and cross-system look-ahead approach was missing to solve this problem. Therefore, it was necessary to develop an approach that takes into account the various aspects of the mobile feature integration from a generic perspective. Only such an innovative and holistic approach provides the possibility to integrate mobile features for all the product lifecycle phases and to avoid isolated solutions in form of individual native implementation of mobile PLM applications.

The objective of this research was the modeling of a framework for the integration of mobile features in PLM environments. The identified gaps of the prior art have been used to formulate the unmet needs in requirements. These requirements have to be taken into account in the modeling of the framework. Likewise, technological possibilities and standardized techniques had been considered in the process model and the implementation of the framework prototype. The framework has been modeled generically, so that the exchange of framework components and mobile features is feasible and thus a variety of implementations beyond the reference architecture can be conducted.

Chapter 1 describes the present situation and discusses the existing problem as well as determines the study areas and structures for research work. Chapter 2 specifies the requirements of the Mobile Feature Framework for the various dimensions of social, business, and technology. Chapter 3 covers the state of the art based on the specified study areas and describes the existing deficits. Chapter 4 explains the procedures and methodologies of the selected solution approach as well as constitutes the core element of this research work. This chapter describes in detail the individual steps of the Mobile Feature Framework modeling. Chapter 5 deals with the design, modeling, and development of mobile features, which are integrated into the framework in a modular form. Chapters 6 and 7 provide insight into the implementation of the framework prototype as well as the realization and evaluation of scenarios with the specific application examples.

In this research, a framework for the integration of mobile features in the PLM environment was modeled and is dependent on a component-based structure. For this purpose, the components were divided into the application domains of mobile device-specific features, the integration of PDM systems, mobile applications, and the core of the framework. The approach has been presented for all of these components and has shown how they have been modeled, specified, and implemented. Likewise, it was considered that they meet the respective specified requirements and standardized technologies that have been preferably treated for implementation. The framework has been implemented by a prototype, which is described in Chapter 6. The implemented fictitious sample applications follow the specified conditions of the Mobile Feature Framework. The core of this research represents the flexible integration of mobile features in an existing PLM landscape. In so doing, mobile features can be reused without additional integration efforts for future PLM scenarios. The modeled framework can, in principle, basically be applied to other professional domains such as CRM, ERP, SCM, and SLM. The framework component for integration of PDM systems (MDI) can be replaced, for example, by an ERP module or other integrable software systems. Likewise, the applicability to other classes of devices is given, so long as the necessary requirements of the framework are fulfilled. The realization of mobile PLM applications can be done by various formal languages, because the framework early favored approaches of standardized technologies with cross-system capabilities in the entire modeling phase.

8.2 Limitations

The modeled framework defines a structure which is used to integrate mobile features into (mobile) PLM applications. Various aspects such as stability, performance, and security are considered. The object-oriented and component-based framework supports the programmer in the integration of mobile features, but it can not guarantee the quality of mobile PLM applications in IT projects. The successful implementation highly depends on the conceptual approach of the mobile

PLM application. Architectural errors in design can have a serious impact on the implementation. The software quality is highly dependent on the practical experience and professional qualifications of the project members. Strong social skills and teamwork represents a key element of successful project work.

The framework is built on a component-based architecture. The primary advantage is that the framework can be adapted flexibly to the requirements of the infrastructure. Therefore, the components for the Mobile Device Client including the mobile features and system integration component are interchangeable. However, these advantages are tied with high demands to the individual components. The developer must know the interface specification in order to realize generic components for further mobile device classes and system components. In the short term negative effects might be expected, because a certain training period into the framework and the use of interfaces are essential to meet the individual requirements of the application. The advantages for the developer unfold only in medium and long term with an increasing number of implemented mobile PLM applications through the reusability of components. The fact that mobile PLM applications access the interfaces of the component-based framework architecture, additional effort of time and cost for application-specific developments can be avoided.

Many business units consist of complex and cross-process operations which mostly require the interaction of different responsibilities. Various IT systems and infrastructure components are used for managing, exchanging, and processing PLM information. In order to expand or redesign individual process components consistent with mobile features, all software systems must be technologically and organizationally integrated as a module in the Mobile Feature Framework if the system has the capacity for integration. In the past, the integration was done mostly by individual approaches for each phase of the product lifecycle between sub-processes and mobile IT applications. However, this procedure has significant drawbacks. With the increasing number of components to be integrated, simultaneously the integration effort increases exponentially. Changes to individual process steps or mobile IT applications result in multiple adaptation iterations. Therefore, the individual and isolated integration of many application scenarios is uneconomical from the business perspective. In this case, the framework shows its strengths by the uniform integration approach. However, the technological progress also requires a continuous development of the framework components and mobile features. Serious changes can be observed particularly in the mobile sector, because new device classes are pushed in the market in short time intervals and new technological standards are being set. Mobile platforms are strongly affected because the support is completely discontinued after a few years. The missing technological adaptation of implemented framework components to new technologies would ultimately lead to unsupported components (e.g. communication protocols) in newer software versions and incompatibilities may result in various infrastructure components. However, the business logic of the modeled domain model would not be affected, because the layered architecture of the business logic is isolated from the infrastructure, which means that each layer operates independently.

8.3 Outlook

The increasing need for mobility encourages the technological progress of novel and innovative mobile products and services in the industrial sector. Mobile devices represent for PLM information an important endpoint in order to consume and contribute information in mobile situations. Thus, they are of fundamental importance for the product lifecycle and will constitute a central and integral building block of modern mobile PLM activities in the future. Therefore, mobile devices need to be understood as a tool of the product lifecycle management, rather than considered as an isolated system solution. Currently, the young mobile PLM market is still at the beginning of a rapid development in which the potential of mobile technical and organizational possibility has not yet been recognized and exploited to the full extent. In order to promote the development of the mobile PLM market, it must be possible to develop sustainable mobility solutions with standardized access and communication methods to justify and protect the intensive investments in the PLM infrastructure. This applies in particular for the research area of *Wearable Computing* in order to perform future PLM tasks with the support of portable computer systems. However, the use of such devices requires full support of mobile users to accept new technology when they provide clear benefits in daily business operations.

One of the challenges will be to learn and become familiar with new technologies and break down barriers, especially for experienced older generations. For this purpose, it is mandatory to address concerns about subjects like security, privacy, and reliability as well as satisfy the individual needs. The so-called Facebook generation that has grown up in a digital world is encouraged to explore the benefits of new technologies for personal usage. This applies not only to social networks, but also to the intelligent use of mobile features. This discrepancy shows that research for this topic is urgently needed in order to prevent that older generations lose touch with new technological advances. There will be a demand for research of the following fields with regard to mobile features:

- Identification and consideration of generational differences in the industrial environment
- Examination of cultural aspects in approach of product development
- Requirements analysis of mobile features with regard to user roles in the PLM environment
- Investigation and identification of PLM processes that can be suitable for mobile use cases (e.g. no continuous availability of services, acceptance of high response times)

A change from products and information to processes and advanced user experiences become increasingly apparent in the PLM industry. In first place, companies would not like to deal with software products, but they are interested in meeting the demands with corresponding component-based process solutions for daily operations. The altered behavior of customers is increasingly reflected in PLM software solutions vendors. This has the consequence that products must be integrated more intensively with one another to ensure consistent and seamless processes involving all stakeholders without manual intervention during transitions between sub-processes (process breakdowns). Such processes comprise several jointly-acting components with different technical interfaces. The dashboarding has been proven in business for decision-makers as such a cross-component solution. For this purpose, information is retrieved by different complex application systems and visually presented by widgets. Users can thereby present information according to their individual needs. A huge advantage is that complex application systems must not be adjusted by additional customization, since the combination of information takes place in widgets. This approach thus supports the young trend of "de-customization" in which the complexity of application systems is reduced.

The challenge for the Mobile Feature Framework and scope of mobile applications arise in the linking of closely integrated process under consideration of mobile activities. This means that by shifting from products to processes, workflows have to be adapted for mobile activities rather than considering activities isolated of mobile information workers. Therefore, it is necessary to understand the possibilities and capabilities of mobile users in order to align processes according to the user groups. Moreover, the framework supports the de-customization approach of application systems, since the integration of mobile features normally does not require any modifications of the respective data models. However, for some use cases, it is necessary to extend the data model of application systems for some features (e.g. storing the location data of objects). The object recognition by geometric information can reuse existing data. Further approaches of the object recognition by the analysis of surface structures and noises require the construction of libraries. For example, patterns of surfaces and engine noise must be captured, structured, and linked to a corresponding logic. In this subject area, there is still considerable need for further research concerning the interaction between the recognition and identification of objects. Previous approaches and methods are unsuitable for PLM, because the PLM sphere is characterized by completely different conditions and demands.

This research work has presented an approach of integration possibilities to incorporate mobile features for the PLM environment. Now, the presented integration approach needs to be implemented in industrial projects. The prototype of the Mobile Feature Framework serves as a reference implementation for future PLM projects in the mobile environment. The author would like to motivate PLM vendors to consider the integration of mobile users and mobile features strengthen in PLM concepts in order to exploit currently untapped potential of mobile users. Companies and employees are able to benefit to the same extent at different levels of competence.

It is the author's hope that this publication has made a significant and valuable contribution to the research in the field of *Mobile PLM* and will continue to be a valuable resource for enterprises which are connected to PLM network environment, since mobility brings the biggest advantage of being anywhere and everywhere.

Appendix

A Mobile User Culture by Dimensions

This section examines the behavior of mobile users based on the dimensions: Political conditions, financial and economic conditions, formal education, generation, social relationships, health aspects, and physical environment. Therefore, various user point of views as well as aspects of device usage are considered and discussed.

A.1 Political Conditions

During the time period of the Arab Spring[1], the political events changed the communication behavior of the people in this region. Due to the vast shutdown of access to the Internet[2], people were forced to find alternative ways to communicate. Mobile phones play a crucial role in the revolution, because mobile features of the devices were used to share text-based messages with photos and videos on social networks and video platforms. Over micro-blogging services such as Twitter, a new form of communication that has previously received little attention has been established and distributed within few weeks. This type of communication supported activists to organize protests and to motivate people to counteract security controls and thus avoid imprisonment [Cf.

[1] The Arab Spring (also known as Arabellion [Cf. Hor11]) began in December 2010 with several protest movements and uprisings in the Arab world and led finally to the collapse authoritarian ruling regimes.

[2] Over 93 percent of Egyptian networks were disabled [Cf. Cow11]

Bou11, p. 3], which enables them to report to the outside world as well as stay in touch with friends [Cf. How11].

A.2 Financial and Economic Conditions

In some countries, the wide range of smartphones or feature phones is not so prevalent as in Europe or the United States. Although Africa is the world second most connected region by mobile subscribers [Cf. Ree11], this region mostly uses low-end mobile phones from Nokia, since smartphones are still too expensive. This is primarily based on the fact that in some African countries, mobile costs are high and there was a need to reduce costs led to the development of a culture of *flashing* and *beeping* where phone users call a mobile phone number and immediately hang up before the called person can receive the call. Rasirsa a jeweler in Rwanda said:

> "[...] people are not rich enough to call every time. That is why they beep you. Many people only have access cards. When they want to call someone, they go to a public kiosk or use their mobile to beep someone...it is cheaper." [Don05]

Hence, different sets of conditions are composed of response times or phone ring length made in different calling-context. For example, the following interpretations apply:

> "I'm done with my work, pick me up." and "I'm thinking about you." [Don05]

The penetration of mobile phones in Africa is expected to reach over 87 percent by 2016, which represents one of the lowest in the world [Cf. Rus11]. The majority of Africans go online via their mobile phone instead of using desktop or laptop devices, but data-enabled phones are less than 20 percent of the total mobile phone market [Cf. Man13]. However, the number of smartphones is increasing steadily in Africa because the relatively high prices have dropped across the board to an affordable level [Cf. OB12]. In western regions, the people who own a holiday home tend to provide more the mobile number instead of the landline number for accessibility in general [Cf. The09]. In Japan, mobile phones are heavily subsidized in such a way that the resale value is nearly zero. Therefore, it is not unusual to find in the gutter old equipment that contains personal information of the owner [Cf. The09]. In the emerging markets, it is common to use mobile phones with dual SIM to take advantage of cheaper deals for voice telephony and data services [Cf. Can12]. The dual SIM feature is practical and economical for travelers who need to be between various provinces due to the implementation of the national roaming system in China [Cf. Chu12, p. 74].

A.3 Formal Education

Through new mobile features such as location-based services and new forms of media such as social networks, the way of user communications has changed. New technologies enable companies to provide an enhanced user experience for mobile users through tailored views and to address the individual user needs. The associated ever increasing complexity of mobile devices and mobile services created a desire for intuitive operation. Unfortunately, the desire remains often unfulfilled, because the technical features and functions as well as varying interaction techniques need to be conveyed to the user. The imparting of knowledge can be achieved, for example, by instructions or trainings. Whether to use and read manuals or not depends on factors such as the complexity of user guides, the prevailing acceptance of instructions in the respective region, age and gender of the user and the frequency usage of a device. Figure A.1 shows various learning techniques of users. The personal education of each individual user plays a central role in case that mobile applications are not self-explanatory. Thereby, users can be divided into three groups: (A) beginners, (B) advanced users and (C) experts.

- (A) Beginners are users who build a first contact with a specific mobile device. They have no or minimal experience in device handling or operation. Therefore, this group of users asks clearly more questions than it is in the case of advanced user groups. The motivation for these users is the ability to solve a problem at all, even if the optimal solution path is not found and therefore

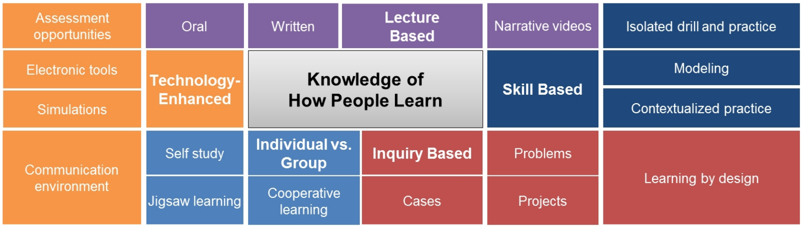


Figure A.1: Knowledge techniques to accomplish specific goals (adapted from [Cf. BBC00, p. 22])

significantly more time is necessary. A lack of support (e.g. work colleagues, friends or training program) would increase the risk that the motivation decreases in finding a solution. A possible solution method of beginners is *try-and-error* to force a solution without having the necessity of the technical understanding.

- (B) Advanced users can benefit from the knowledge of other domains towards beginners. Pattern can be applied to support the current context due to the technical understanding of the other domains (e.g. laptops or previous used mobile devices). Standard functions can be easily adapted from other device classes which help to perform the user tasks. In contrast, device-specific functions are difficult to use without having the technical knowledge and require a certain time to carry out the specific user task (e.g. additional time for reading handbooks and participation on trainings). The motivation to cope with a task on a mobile device with the optimal approach is secondary.
- (C) Experts represent the group with the most extensive expertise. They have knowledge about basic functions and device-specific functions as well as mastered in their practical application. The knowledge is usually not only acquired by training courses or user manuals but also based on years of professional experiences. The accumulation of experience is controlled by a variety of factors such as motivation to find solutions, technical interest, and the current context. The knowledge can be derived from different domains. Experts follow a systematic approach in carrying out a task. Any errors that occur are categorized and appropriate use case scenarios

are derived. Based on the use case, appropriate solutions are created for the problem.

The behavior of mobile users in the performance of tasks was examined in different studies. Oulasvirta et al. [OWAE11] examines all three user groups on their experience and expertise in carrying out of device-specific tasks such as calendar entries, setting up network connections, audio and video recordings. The study results showed that occasional users learned mainly through the routine use of smartphones, while expert users learned more regularly and systematically [Cf. OWAE11, p. 165].

A.4 Generation

Compared with the generation of the early days of mobile communication, the capabilities of the mobile phones and the communication behavior of the users have evolved. While calls were made and text messages by SMS were written several years ago, today, increasingly multimedia capabilities of the mobile phone are used for communications. Experiences and all kinds of contents are not only shared on social networks, but also assessed, commented, and downloaded. Different communication technologies such as microblogging services[3] allow the distribution of content in almost real-time. The communication speed, the way of sharing information, and language of mobile users have changed. Information is exchanged in shorter time intervals and partially in compressed form. Moreover, these changes are applied from different linguistic forms. Through the use of short forms, the user can accommodate more information on a specific context while taking the limitations of the communication medium into account. Likewise, the ease to enter the message plays a decisive role and the aim is to enable the opportunity that the transmitted information can be faster absorbed by the recipient. Communication media has different limitations. Thus, the SMS has a text length limit of 160 characters, whereas the blogging service Twitter has a limitation up to 140 characters per message [Cf. Twi12]. The interpretation of short forms in a message is only possible for

[3] Twitter is one of the most famous microblogging services world-wide. The service was established in 2006 and counted about 1.4 billion members in May 2013 [Cf. two13].

a circle of persons who share the same context with the corresponding background knowledge.

Acronyms are used to form a short word of multiple words. Therefore, the respective initial letter of a word is used. The acronyms can be divided into two areas of commonly used: acronyms and the subject-specific acronyms. General Acronyms are often used in small talk conversations to describe for example actions or emotions. Subject-specific acronyms can often be understood only by a restricted group of people with the required expertise. The specialization goes so far that a lot of specialized acronyms are used and understood only in certain departments of a company. Large companies such as IBM provides a company-related dictionary for their employees. An ellipse is understood as the omission of words in order to shorten the sentence length. Ellipses obtain their meaning from their context [Cf. Joh08, p. 1]. Through the linguistic or situational context, the meaning of the sentence can be reconstructed [Cf. Glü10, p. 173]. However, there are risks for participants in the communication during context changes when a long time interruption occurs, so that contextual relationships cannot be correctly recognized and understood.

Ellipses are frequently used in the spoken language and becoming increasingly common for applications in the electronic communications out of the mobile situations. Therefore, communication mediums such as SMS, instant messaging, or microblogging services are used for ellipses and in the case of SMS, it has a dialogic character, which allows participants (or sender) to use a plurality of word-savings without compromising the interpretation of the reader [Cf. Sch12, p. 90]. The following examples demonstrate two different types of ellipses.

Examples of the elliptical type: Gap formation

1. *Should I call you, or* [*should*] *you* [*call*] *me?*
2. *I expect you to help the design department, and you* [*expect*] *me* [*to help*].
3. *Michael has done the engineering work, and Luis* [*has done the work*]*, too.*

Examples of the ellipses type: Stripping

1. *Should I do it, or* [*should*] *you* [*do it*]*?*

2. *Max asked the customer to stay, and*
   [*Max asked*] *the employee* [*to stay*] *too.*
3. *Julia works at Heidelberg, and Michael* [*works at Heidelberg*] *too.*

Ellipses also occur in other languages, such as the following German example shows:

[*Ich*] *bin noch in der Produktionshalle,* [*ich*] *komme zum Meeting in 5 Min.*

The following ellipses examples show how many characters a mobile user can save when entering the message. By saving time, the message can be quickly delivered to the recipient and thus promoting a more fluid dialogue (or conversation).

Message without ellipses, abbreviation, and liquidation:
Guten morgen Hans, ich habe gestern mit dem Produktionleiter gesprochen. Ich denke dass wir mit dem aktuellen Design weitermachen können. Ich werde dir heute Nachmittag die weiteren Schritte erläutern. Viele Grüße Michael

Message with ellipses, abbreviation, and liquidation:
Guten morgen,habe gestern mit dem PL gesprochen. Denk das wir mit dem akt. Design weitermachen können. Werd dir heute Nachmittag weitere Schritte erläutern. Grüße Michael

Through liquidation, the message could be reduced from 224 characters to 171 characters. Thereby, 53 characters have been saved and this represent a reduction of approximately 31 percent. The resulting time saving depends on the input speed of each mobile user and input assistance tools such as T9[4] or Swype[5].

[4] Text on 9 keys (T9) is a method for simplified text entry which will display a list of word suggestions by entering the first few letters in which the user can take over the word choice.

[5] Swype is a method to input text by swiping the finger or with a stylus across the soft keyboard. Swype is proprietary and supports operating systems such as Android, Windows Mobile. However, not all languages are supported.

A.5 Social Relationships

In this context, social relationship is understood as a set of social interactions through mobile devices to others humans and the social interaction with surrounding humans. Both types of interactions are related to the mobile user and may be in competition to the mobile user's attention. A social relationship always consists of at least two or more participants who can have an impact on the relationship during the social interaction. Therefore, a balance of social relations is an essential requirement of mobile users in order to create successful social links between other persons.

In Paris, people feel quickly disturbed and complain when intimate matters are discussed loudly on the phone. In contrast, rarely registered complaints were made in London [Cf. The09]. In both cities, the people tended to separate conversations between face-to-face and telephone. In contrast to these both cities, customers in Madrid tend to integrate people in the telephone conversation standing outside [Cf. The09], because in Spain most of the social life takes place on the road than in other countries, such as in the UK or Germany [Cf. Hol07, p. 93]. Spanish people prefer answering calls and turn off the voicemail. As a consequence, Spanish people consider unanswered calls as impolite. This could explain the strong sense of social obligation towards friends and family [Cf. The09]. The Italians and Spaniards tend to answer calls in restaurants during business meetings, conferences, and even sometimes during a concert. The discreet writing of text messages or instant messaging under the table is also a commonplace during business meetings. In India and in parts of Africa, it is common that people take phone calls during a theater performance, even at very formal events or speeches. However, in Japan it is customary to completely turn off the mobile phone on public events because the light of the screen might annoy other persons. In contrast, in parts of Africa the people are even proud to present his own device and show incoming international calls to other people [Cf. Can12]. Pethiyagoda from St. Anthony's College, Oxford, said:

> "Indian society has a long tradition of tolerance, including in terms of allowing others to infringe on what those in the West would consider one's personal space." [Can12]

The Chinese often let themselves be interrupted in a conversation or meeting, fearing that they could lost a business opportunity. Uzbeks use their mobile phone only rarely in public, since the police could potentially eavesdrop on the conversations [Cf. The09].

In some countries it is a common practice to have multiple mobile phones, because this indicates the degree of importance. In Germany it is common to use a mobile phone for personal use and a second mobile phone exclusively for business purposes. In Latin America some managers even use a dedicated mobile phone only for communication with their superiors [Cf. The09]. In a study conducted by Synovate was found out that 72 percent of Americans feel that loud conversations in public in the worst habits of mobile users [Cf. Can12]. Since Apple's introduction of Siri[6], another example of a conditioned habit of mobile users seem to be established. The repetition of simple questions can be required by a variety of reasons which makes surrounding people angry; this service is provided by communication assistant. All these examples above give an insight into the diversity of social aspects of a relationship. The social interactions can take an influence on the social relationship of people. The nature of the social interactions and its impact on the social relationship may have country-specific or region-specific reasons and will not be further examined in this work.

A.6 Health Aspects

The health aspects describe the direct and indirect effects through the use of mobile devices. In Japan, it is frowned upon in public (e.g. public transport, classrooms and restaurants) to hold talks on the phone [Cf. IOM05, p. 11]. Therefore, numerous signs and announcements in Japan's public transport indicate not to use the mobile phone (see Figures A.2a,

[6] Speech Interpretation and Recognition Interface (Siri) was introduced as a software component in products such as Apple's iPhone and iPad in 2007 and is used to process naturally spoken language in order to perform personalized functions.

A.2b, A.2c, and A.2d). People who ignore this rule, can be personally addressed by security personnel that the mobile phone use is not allowed and this is to be avoided [Cf. Kei13]. One reason for this is that people with a pacemaker continually increase with the ever-aging population of Japan[7]. In narrow spaces it might be possible that the radio waves of the mobile phone may potentially affect the function of the pacemaker [Cf. ETI^{+}11]. Since 70 percent of Japanese carry their mobile phone within a meter over the whole day [Cf. HG11, p. 92], the social concern in the minds of the people came up. Users have relatively long commuting distances on public transport to the work office and limitations in the use of mobile devices , therefore having less available time. Thus, the need

Figure A.2: Mobile phone sign in Japans trains (adapted from a) and d) [Cf. Tea10], b) [Cf. Pou10], and c) [Cf. Yuj11])

[7] The Japan population aged 65 years old and over increased by 13.9 percent based on the population count from census in 2010. [Cf. Sta11, p. 150]

awakened to maintain social relationships in mobile situations leads to the growth in demand for mobile data services and services such as *i-mode* were created to satisfy the user needs. The mobile Internet service i-mode was launched in 1999 in Japan by NTT Docomo[8] and allows mobile web access, e-mail communication, and other data services such as news and weather forecasts. Thus, the i-mode service is in direct competition to WAP. In European countries, there is widespread concern about potential health risks of electromagnetic fields. A survey in 2010 commissioned by the European Commission showed that 46 percent (versus 48 percent compared to 2007 [Cf. Eur07]) of EU citizens assess the health risks of EMF caused by mobile phones with "*Very concerned*" and "*fairly concerned*" (see Figure A.3) [Cf. Eur10, p. 68]. Thereby, the main concern has been expressed by Italy and Greece with 81 percent (+12 percent and -5 percent compared to 2007) and Cyprus with 80 percent (-2 percent compared to 2007). The formation of the people was also a crucial factor, because 48 percent of respondents with a lower education level (versus 63 percent compared to 2007) and over 67 percent of educated persons (versus 76 percent compared to 2007) with an age of about 20 years believe that mobile phones are the source of EMF [Cf. Eur10, p. 49]. This view is shared by a wide variation of occupational groups. Such fears can strongly influence the intensity and nature of use of mobile devices. In 2011, the World Health Organization's International Agency for Research on Cancer (IARC) has classified the radiofrequency electromagnetic fields as "*possibly carcinogenic to humans*" [Cf. Wor11b, p. 1].

> IARC director Christopher Wild said: "[I]t is important that additional research be conducted into the long-term, heavy use of mobile phones. Pending the availability of such information, it is important to take pragmatic measures to reduce exposure such as hands-free devices or texting." [Wor11b, p. 2]

[8] NTT Docomo is the largest mobile operator in Japan with more than 60 million customers [Cf. NTT13]

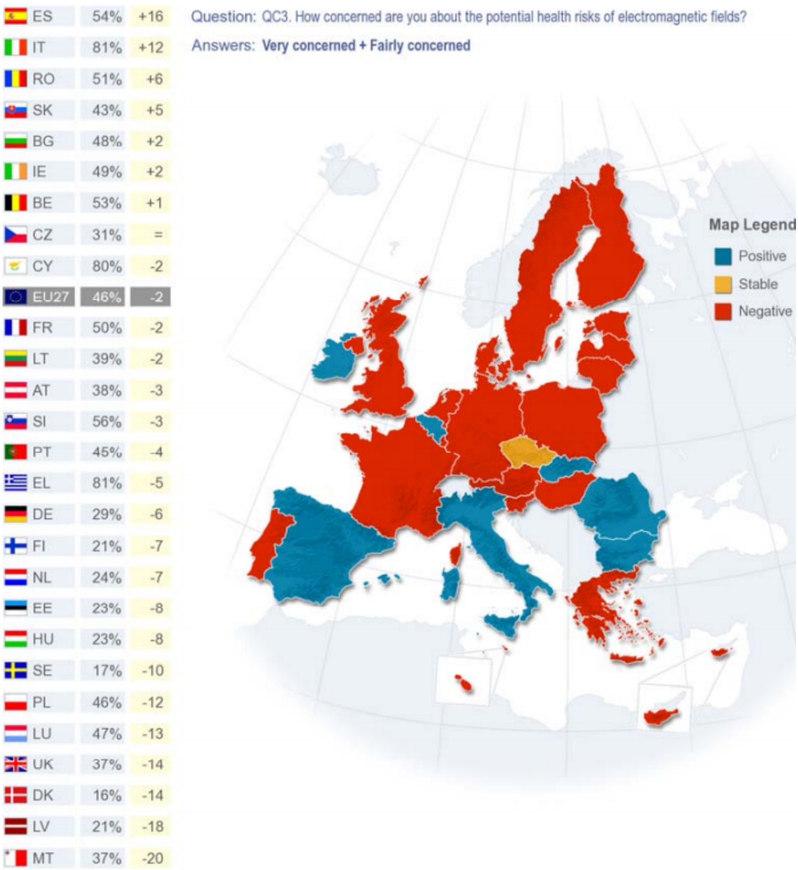
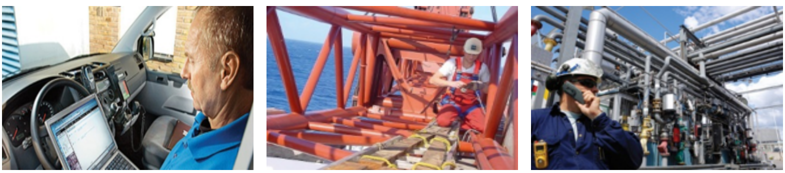


Figure A.3: Health concerns about mobile usage in the EU (adapted from [Cf. Eur10, p. 68])

Given that information from long-term studies for the intensive use of mobile phones is not available, call tracking apps like "Track phone radiation" [Cf. Taw13] has been developed to help users in identifying the radiation exposure value. The app detects and analyzes the mobile device and network data, calculates the Phone Radiation Level (PRL) and thus the exposure of the mobile user. Once the limit has been exceeded, the user is informed and suggestions to reduce exposure are provided by the mobile app. Moreover, the app is able to direct the user to a better reception area in order to reduce the transmission power of the mobile phone and thus also the exposure.

A.7 Physical Environment

The physical environment has an influence on the type of mobile device and the mobile user. Mobile devices take our physical capabilities and our attention [Cf. KL99]. The physical abilities of humans are represented by eyes for the visual processing of information and the hands used to hold the device as well as to operate for example through gestures. The attention of the user will be spent on the interaction with the mobile device in order to carry out tasks in a mobile context. To deal with such mobile contexts, certain predetermined conditions have been met. For example, the driver would have to stop a car first to apply the visual attention required for the mobile device (see Figure A.4a) in order to successfully interact with the device. This often does not happen and consequently leads to car accidents. A service technician must first find a quiet working position before he can carry out the interaction with the mobile device (see Figure A.4b). External noise sources such as low light conditions and high ambient noise can also influence the interaction with the mobile device negatively (see Figure A.4c). These influencing factors exist for a variety of other professional groups in



a) b) c)

Figure A.4: Physical environment (adapted from a) [Cf. ACL07], b) [Cf. TTS14], and c) [Cf. Nas12]) the business environment. In contrast to car driver, disproportionate speakers for mobile devices are used in agriculture by farmers to improve the acoustically understanding in a communication between partners while working on loud tractors [Cf. The09]. Even climates can influence the mobile user behaviors. In mega-cities like Mumbai which is located in the tropical climate zone, it is common that the menfolk have rarely wear jackets to stow their mobile device. Instead of Jackets, most people use the shirt pockets for stowing objects that offer a limited space for smartphones (see Figure A.5) [Cf. The09]. The communication will

Figure A.5: Shirt pockets as holder for mobile phones in India (adapted from [Cf. Gar13]) be limited in regions without mobile network coverage and without data services. In addition to the technical ability to access the Internet, the cost will also be a deciding role in the environment of the mobile user. Excessive costs would represent a barrier for mobile data usage. According to the ITU[9], the number of mobile broadband connections increased from 268 million in 2007 to 2.1 billion in 2013, which reflects an average annual growth rate of 40 percent [Cf. Int13, p. 6]. Particularly in Africa, a strong growth in the mobile sector can be observed, because mobile data services enhance the quality of human life. For example, SMS service is used for the detection of counterfeit drugs to verify the unique codes on pharmaceutical packages [Cf. OB12].

[9] International Telecommunication Union (ITU)

B Mobile Technology

B.1 Framework Requirements

Property	Wired Networks (LAN)	Mobile Networks
Connection delay (Socket open & close)	Normal: less than 0.2 second; Max: assume 5 or 10 seconds is failure	Normal: 2 to 5 seconds; Max: must wait 30 to 60 seconds before assuming failure
Response delay	Normal: less than 0.2 second; Max: assume 1 or 2 seconds is failure	Normal: 1 to 3 seconds; Max: 30 seconds before assuming failure
Idle TCP[1] sockets	TCP sockets can sit idle indefinitely; limited only by application protocol expectations	Varies, with interference; they may stop working without either end seeing a close, abort or reset
UDP[2] reliability	On modern 100Mbps switched Ethernet, UDP/IP[3] is reliable with packet loss rare	UDP packets loss is to be expected (depending on location, weather, network congestion and other factors)
Costs to communicate	Only cost of generating network messages is the impact on other devices and communications	Costs based on max. expected data bytes per month; every message sent potentially costs money

Table B.1: Comparison between stationary and mobile networks [Cf. Dig09, p. 7]

B.2 Mobile Network Types

Network Type
Wireless Personal Areas Network (WPAN) - 0,2 to 50 m
- FIR-IrDA with 4 Mbit/s and VFIR-IrDA up to 16 Mbit/s - Bluetooth up to 3 Mbit/s - IEEE 802.11g-Standard up to 54 Mbit/s
Wireless Local Area Network (WLAN) - up to 300 Mbit/s (IEEE 802.11ac)
- IEEE 802.11h, data rate up to 54 Mbit/s - IEEE 802.11n, data rate up to 600 Mbit/s - IEEE 802.11ac, data rate at least 1,3 Gbit/s up to 7 Gbit/s (Draft version)
Wireless Metropolitan Area Network (WMAN) mainly constructed by Standard IEEE 802.11
Wireless Wide Area Network (WWAN)
- Universal Mobile Telecommunications System (UMTS) - up to 42 Mbit/s with HSPA+, otherwise max. 384 kbit/s - Global System for Mobile Communications (GSM) - up to 220 kbit/s with EDGE, otherwise max. 55 kbit/s for GPRS - Long Term Evolution (LTE) - up to 300 Mbit/s

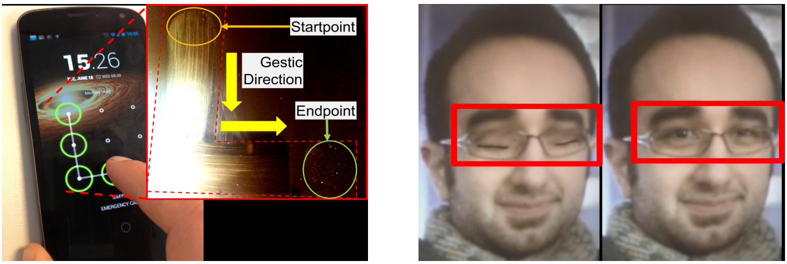
Table B.2: Network types

B.3 Mobile Device - Authentication Issue

Gestures for the authentication of devices are used in applications to provide an alternative input method for entering complex passwords using the on-screen keyboard. For example, Android provides a method to unblock the screen by drawing a previous defined pattern (screen lock

[13] Transmission Control Protocol (TCP)
[14] User Datagram Protocol (UDP)
[15] Collection of communications protocols used for various networks

pattern). From a safety aspect, this method is unsure because visible traces of hand perspiration lead to a quickly guess of the correct pattern (see Figure B.1a) [Cf. AGM$^+$10]. A further gestural method allows an unlock of the screen using the face (see Figure B.1b). Vulnerabilities have also been identified in this method, because the screen can be unlocked through image manipulation [Cf. Kel13]. In the future, more security could be achieved through the detection of human grimaces (Patent by Sipe [SSN13]) because this method provides a higher degree of individuality by detecting, for example, motions of the eyebrows, smiling, or sticking out the tongue.



a) *Unlock - Pattern* **b)** *Face Unlock*

Figure B.1: Security issues with gestics, (adapted from b) [Cf. Mus12])

B.4 Gesture Types of Mobile Users

Fulton [VWW11] describes in a reference card 13x total core gestures (see Figure B.2), 3x Basic actions (e.g. Select and Open), 17x Object-related actions (such as Move, Rotate, Scale up/down), 22x Navigation actions (e.g. Adjust view and Scroll), 5x Drawing actions (e.g. Cut, Reject and Undo), and other platform-specific gestures[4]. Core gestures that are platform-independent represent fundamental movements. However, the implementation of such gestures techniques in mobile applications is a challenge, because gestures represent communicative movements

[4] 7x Android gestures, 8x Apple iOS gestures and 7x Windows Phone gestures

of the mobile user and therefore can be drawn and applied differently. Physical conditions (e.g. finger length, finger size and finger nail), the speed of the gesture drawing, and the manner of carrying out (e.g. Short Draw, variation of the finger pressure, inaccurate gesture reproduction) can seriously influence the detection and interpretation of the gesture by software. Figure B.3a shows that the size of the contact field on the touch screen depends on the size of the user's extent fingertips.

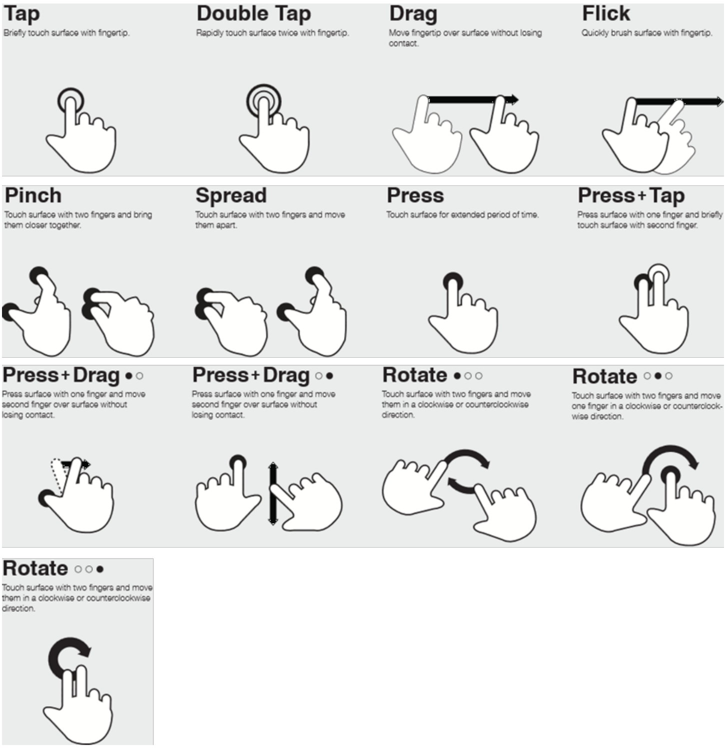
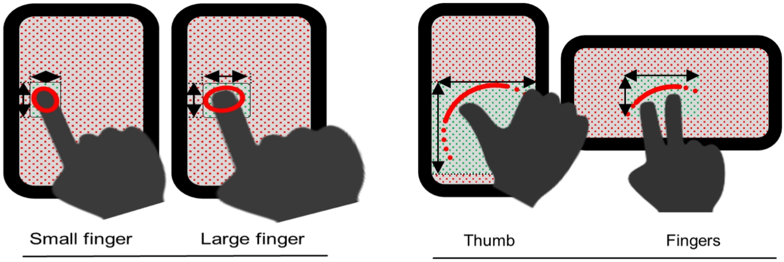


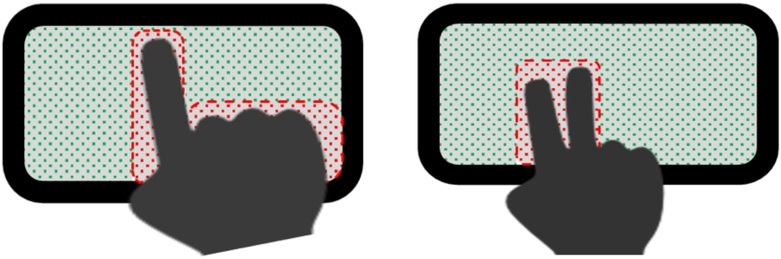
Figure B.2: Touch core gestures (adapted from [Cf. VWW11])



a) *Contact field* **b)** *Radius*

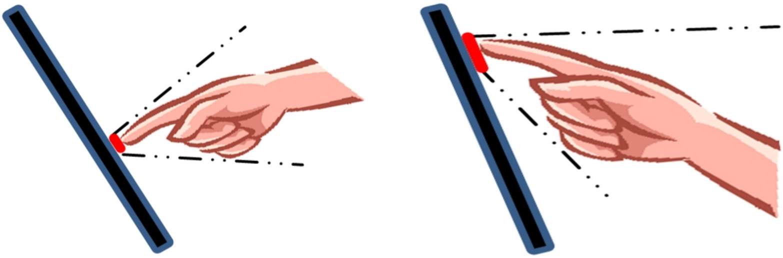
Figure B.3: Touch gesture issues

An extent fingertips would cause that the gesture recognition software would interpret an excessive larger contact field and this would lead that multiple elements are selected in the mobile application (e.g. check boxes in a form). Figure B.3b illustrates the limited movement of the thumb and the spreading of the fingers. Application elements that are located outside the green area cannot be easily reached by the user and thus negatively affect the user experience. Prevailing climatological conditions such as frost cause that mobile users must wear gloves and this limits the flexibility of hand gestures. Therefore, designers should take special care in the design of the surface, so that the arrangement of the elements in the mobile application is taken into account and a certain tolerance for gesture recognition is considered to avoid a negative user experience. In addition to physical constraints, visual influences may also play a role in the interaction with the mobile device. Depending on the angle of the gesture which is carried out, different areas of the display can be hidden and thus information cannot be visually perceived for a short period of time by users. Arrayed elements in the lower viewing area represent the largest overlapping field by fingers (see Figure B.4a and B.4b). The inclination angle of the mobile device and the hand are crucial for the size of the contact field on the display of the mobile device. If the finger contact to display is in a right angle (90 °), then the contact point is the smallest (see Figure B.5a). In contrast, the circumference of contact point is growing with a larger degree of angle (see Figure B.5b).



a) *Single finger* **b)** *Dual finger*

Figure B.4: Touch gesture concealed areas



a) *90° / lower contact* **b)** *= 90° / larger contact*

Figure B.5: Touch contact surface area

B.5 Example of a User Activity Profile

Figure B.6 shows schematically two activity profiles that present an optimal day routine with multiple (mobile) devices and a sub-optimal daily routine using only a stationary device. Depending on the user-specific characteristic of the dimensions, the activity profiles of users can vary significantly. The day schedule is characterized by different activities. The use of mobile devices in continuous time intervals over the day allows user 1 a more flexible coordination of new upcoming activities. In the schematically day, schedule of user 1 is shown that they use the mobile device before starting the work day to get an overview of the activities of the day so that they are able to consider changes in the day routine. While user 1 is already prepared when arriving the office and is able to directly perform tasks on the workstation, the user 2 is busy to organize his day routine. This is followed by a coffee break which is used for the exchange of new information and ideas. The mobile device of the user 1 supports this situation by providing access to information that is relevant for the discussion of a new idea. User 2 memorizes the idea with the appropriate comments from the discussion to perform further activities (e.g. query background information) on the stationary workstation. User 1 has all the necessary information already collected to address the idea in a meeting, while user 2 has to invite the participants to a separate meeting. During the lunch break, user 1 is able to inform a colleague via instant message using the mobile device, so that they can support him in the discussion about the current progress of the project in the management meeting. The colleague was promptly informed and this allows him to prepare some information before the meeting starts.

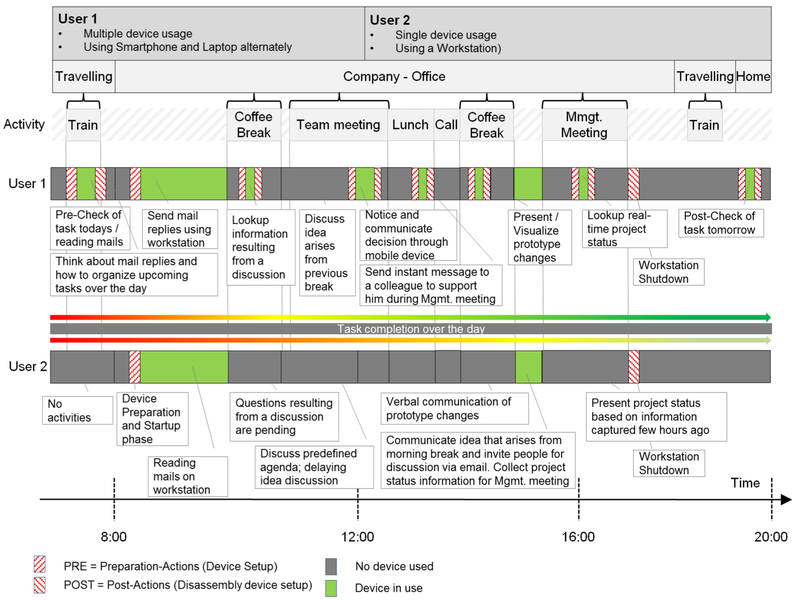


Figure B.6: User activity profile over the day

User 2 cannot reach his colleague shortly before the meeting starts, so that further information cannot be presented in the meeting by colleagues. In addition, user 1 is able to present real-time information, while user 2 only presents information which is already few hours old. In the evening hours, User 1 obtains an overview of high priority tasks for the next business day with his mobile device, while user 2 stays uninformed. The colored arrows indicate the progress of completed tasks of the users throughout the work day.

B.6 Mobile Device Classification

Table B.3 lists the different classifications for mobile devices and their respective applications with a short description. This table does not purport to be a complete list, because several new devices and applications are published nearly every day.

Device Types	Industries	Intended Use
Device Classification: Communicating		
Communication	Consumer goods	Access to a complete set of communication vocabulary
Customer feedback	Retail	Data evaluation is transmitted to server for storing and generates various reports
Two-way Radio	All industries	Used for conversations
Information display	All industries	Display any kind of information
RFID[5] Reader	All industries	Used for assets management, logistic management, personnel location management, production line management, high-value commodity anti-counterfeiting

Continued on next page

[5] Radio Frequency Identification (RFID) is a wireless contactless technology that uses electromagnetic waves to transfer data from a transponder (commonly known as tag) which is attached to an object for tracking

Table B.3 – continued from previous page

Device Types	Industries	Intended Use
NFC[6] reader	All industries	Logistics tracking and management system
Infrared reader	All industries	Infrared data acquisition terminal
Device Classification: Processing		
System boards	Semiconductor	Banking, telephone, health insurance cards, luggage tags
Medical (e.g. pacemaker)	Health care	Monitoring of heart's native electrical rhythm
3D measuring	Aerospace, automotive, consumer products, manufacturing, and research	Reverse engineering of geometric entities
Restaurant order	Catering	Manage restaurant orders
Credit card processing	Catering	Transaction processing for card payments in shops, bars, restaurants, etc.
Device Classification: Collecting, Tracking, and Logging		
Delivery information	Logistics and transportation	Manage delivery information and acquisition
Energy management	Consumer goods	Energy monitoring and control of specific devices throughout the home

Continued on next page

[6] Near Field Communication (NFC) is an international communication standard for contactless data exchange over short distances

Table B.3 – continued from previous page

Device Types	Industries	Intended Use
Data logger	Industry, science, and trade	Measure and collect data such as temperature, relative humidity, barometric pressure, or CO_2 concentration. Electronic data loggers are used to monitor environments within structures to control climate-intensive production processes and computer data centers, control panels, wind turbines, and warehouses.
Human tracking	Health care, consumer goods	Positioning, monitoring, safety protection for kids, school students, aged person, company staff, disabled person, pets, asserts
Fingerprint	All industries	Time tracking
Barcode scanner	Logistics and transportation	Scan barcodes
Device Classification: Printing and Scanning		
Printing and scanning	All industries	Scan and print a label or message directly on the package
Handheld thermal printer	All industries	Used for asset management, medical and hospital, cargo inventory, electric power, logistics and transportation, hotel booking, coffee shop booking, and restaurant booking
Device Classification: Monitoring and Controlling		
Analysis and inspection	All industries	Balances rotating machinery; field balancing of rotating machinery in one plane or two planes
Mobile phone anti-theft	Consumer goods	Used for exhibition, supermarket and shops for displaying all kinds of electronic products

Continued on next page

Table B.3 – continued from previous page

Device Types	Industries	Intended Use
Biometric identification	Security	Biometric identification and access control
Device Classification: Measuring		
Networks	Information technology	Troubleshooting and management of computer networks; monitors the general health of a network
Colors	Print media	Programs for the production of characteristic printing curves or colourimetric quality control
Colors	Consumer products, manufacturing	Selecting, measuring, formulating, communicating and matching color
Material testing	Oil and gas industry	Evaluate the pipeline degradation and remaining service life
pH value	Chemical industry	pH measuring instruments
Temperature	Climatology, agricultural meteorology, pharmaceuticals industry	Measure surface temperatures, solids, or liquids
Electrical	Civil and industrial electric plants	Power quality analysis
Medical (Blood glucose meter)	Health care	Measure blood glucose

Table B.3: Mobile device classification

B.7 Mobile Application Characteristics

Characteristics	Remote Display	Web	Hybrid	Nativ
Technical Specifications				
Representation of the user interface	Rendered by local client	Rendered by browser	Rendered by browser	Native UI elements provided by mobile OS
Communication protocol	RDP, ICA, Remote Framebuffer Protocol	HTTP / HTTPS	HTTP / HTTPS for web access	-
Cross development capability	High	High	Middle (Container is platform-specific)	Low
Used languages	Depends on what languages are available on server side, but generally a variety of languages exist	HTML5, CSS, PHP, Java, etc.	HTML5, CSS, PHP, Java for web contents and specific languages of the mobile OS for the container	Depends on languages available on the mobile OS (e.g. Google Android: Java, Apple IOS: Objective-C)
Vendor independence	Depends on protocol	Yes	No, but various frameworks exist	No
Hardware features	Limited access to local hardware functions	Very limited access	Limited access to local hardware functions (Depends on platform and framework)	Full-featured access to hardware functions
Security				
Security concept	Exclusively on server side	Exclusively on server side	Mostly on server side	Locally and on server side
User role concept	Implemented on server side	Implemented on server side	Implemented on server side	Implemented on server side

Continued on next page

Table B.4 – Continued from previous page

Characteristics	Remote Display	Web	Hybrid	Nativ
Blocking of functions	Implemented on server side	Implemented on server side	Web content can be blocked on the server side, local content only partly	Functions are represented by different app versions (e.g. Light and Professional version), but functions can not be locked afterwards
Security updates	Implemented on server side	Implemented on server side	Depends on update (native function forces a new app version)	New app version
Usage				
Installation required	No (only access client must be installed)	No	Yes	Yes
App availability	Network connection always required	Network connection always required	Network connection required under certain conditions (e.g. for updating web content)	Offline available
Bandwidth required to run the app	Low to medium (depends on the protocol)	Medium	Low (varies by content)	No
Performance	Low (screen data)	Medium	Medium (web content)	Very high (native content)
User interface access	Access using the local client application	Access using the browser	Access using the embedded browser	Local application
application execution	On server side	On server side	Container runs locally while web content is processed on server side	Local

Continued on next page

Table B.4 – Continued from previous page

Characteristics	Remote Display	Web	Hybrid	Nativ
Store business data	On server side	On server side	Local and on server side	Local
Distribution	Manual setup	Web	Application distribution platform	Application distribution platform
Ability of customization				
Consideration of the user context	Not possible	Limited	Possible	Possible
Company-specific adaptations (e.g. logo)	Conditional possible (depends on application)	Server-side adaptation	Through new app and server-side adaptation	By providing new apps
Valuation				
Pros	- Data protection is ensured	- Reusability / portability of the source code - Low development costs - Widespread Web technologies - Web content updates can be provided without new app	- Low development costs by outsourcing native functions to web-based functions - Web content updates can be provided without new app	- Full platform capability of the mobile OS (incl. access to hardware features) - Higher performance with native UI - Approval process available (through application distribution platform)

Continued on next page

Table B.4 – Continued from previous page

Characteristics	Remote Display	Web	Hybrid	Nativ
Cons	- Low UI performance - Data and functionality of the app are not offline available - Manual effort for the distribution - Limited access to hardware features	- Low UI performance - Data and functionality of the app are not offline available - No generic approval process available - Very limited access to hardware features (application cache, WebStorage, Web-SQL database) - Different native browser with various support of web standards	- Each mobile platform requires a specially implemented container - Limited access to hardware features - Different native browser with various support of web standards	- Source code is limited reusable for other mobile platforms - Data backup not guaranteed - High development costs caused by platform-specific languages

Table B.4: Characteristics of mobile applications

C PLM and PDM

C.1 PLM Industries

The consideration of appropriate methods for the provision of archiving forms began due to the computerization of construction jobs by moving away from the drawing board to the computer-aided design (CAD). Over the years, the needs for a central archiving possibility for an increasing amount of data (e.g. drawings, documents) reinforced to reorganize file-based directory structures which have been grown over the years [Cf. SW11, p. 25]. These rigid and inflexible organizational structures lead to the development of integrated components (CAx) for PDM systems as well as management components such as workflow management and access rights management. In the past, PDM systems were focused heavily on the construction sector. In recent years, an expansion to other industries that do not pursue the objective of design of a mechanical products was made. In the following sections, these new PLM industries are briefly discussed.

Life Science

Today, companies in life sciences are significantly more under pressure to deliver new drug on the market in shorter product development phases [Cf. Wor11a, p. 117]. This pressure arises, inter alia, because the patent protection for blockbuster drugs expired after a fixed time period[1]. Competitors exploit this to deliver similar drugs on the market and produce many so-called generics at a fraction of the original price. This fact sets the pharmaceutical companies under pressure so that they have to introduce new and modified drugs frequently. The introduction of PLM for life science companies can efficiently adapt their development processes to regulatory requirements and the cost pressure as well as take patients in the early stages of the development into account. This enables companies to increase their market opportunities and ensure the survival in a highly competitive market.

Apparel Industry

During the last years, PLM have made their arrival in the fashion and apparel industry. The textile industry is characterized by globally distributed development, manufacturing, and logistics processes [Cf. Eiß11]. Since the market is calling for two but at least four fashion seasons, some companies present every two weeks new consumer goods and every six new collections [Cf. VK10, p. 56]. Therefore, it is especially important for this industry to design work processes and workflows efficiently to enable an integrated and open collaboration across organizations. The fashion company s.Oliver with over 240 stores takes advantage of a PDM solution with the aim to manage the complexity of the product lines, shorten lead times and improve the global collaboration [Cf. Eiß11]. Moreover, In addition, control and release of collections and materials, the creation of master data and BOMs, the procurement of goods pattern as well as the creation and simulation of online stores to simulate and study the shopping experience take place. This support companies to improve the brand identity and to consider the individual needs and preferences of customers.

[1] Due to long development times, the patent protection for pharmaceuticals is 20 years, but generic drugs can be admitted after 10 to 15 years after the initial registration of the original drug.

Financial Services

PLM has a minor importance in the financial world today. However, PLM can already address the three core themes of the financial world by the following elements [Cf. Ada10, p. 6]:

- **Workflow Management:** Through the coordination of resources in the product development (e.g. derivatives products, savings and loan products), activities can be efficiently controlled and monitored. Responsibilities can be more efficiently managed and work processes can be dynamically adapted to the needs by the integration of workflow and portfolio management. Financial products are also subject to continuous adjustments to the legal framework conditions of regulated markets. Accordingly, financial products are adapted to the market conditions by product redesign, follow-up products, and discontinued products. Any change requires decisions, which must be appropriately documented.
- **Compliance Management:** Due to the banking regulations of Basel I-III[2], banks are required to comply with established rules which must be verifiable. In this respect, the compliance management of PLM solutions can monitor and recorded the activities through detailed audit trails over the entire period. Pinpointed testing and correction options in the specified workflow allow the monitoring of compliance in respect of regulatory requirements.
- **Requirement Management:** The capturing, analysis, and evaluation of customer needs as well as their considerations in the product development of financial products are fundamentally similar to other products. Functional and non-functional requirements are captured for financial engineering products instead of technical products.

[2] Basel refers to a set of rules for bank regulation applied in the Group of the twenty major industrialized and emerging economies (G-20). Basel III was formulated in response to the initiated financial and economic crisis in 2007 [Cf. Deu13].

Consumer Goods and Retail

The retail sector is marked by complex supply chains, multilayered distribution channels, and a variety of procurement processes. Due to the rise of private label products[3], dealers increasingly use PLM solutions in order to collaborate with people and companies more efficiently in the stages of product development to monitor supplier's quality and to organize the procurement management. PLM supports retailers in all aspects of private label products to manage these products, to define and monitor regulatory conditions (e.g. pollution of material) with verifiable elements against the suppliers as well as organize the packaging of products. The suppliers obtain access to specific product information from the contracting authorities via specially adapted PLM portals to confirm quality requirements (e.g. regional defined material specifications) and delivery dates as well as prove compliance with standardized improvement and planning methods. In addition, the integration of native artwork design tools allows designers to create sophisticated designs and styling of 3D models that are managed in a structured manner in the PDM system. By involving people from the product manufacturing, potential manufacturing problems can be detected early and a solution can be found with the support of collaboration tools [Cf. Ray10, p. 101]. Virtual sales rooms allow the simulation of real selling space to analyze the consumer behavior.

Natural Resources

Another important industry is the exploration, planning and extraction of natural resources of the mining, and oil production. PLM supports the exchange of information through a central data repository and collaboration in the planning and visualization of entire geographic

[3] Private label products are defined as independent brands and are customized and produced by manufacturers for dedicated dealers. The dealer undertakes the product sales, advertising and sells the product with its own brand name [Cf. Ray10, p. 102]. The dealer guarantees the product quality with respect to the consumer. Private label products are often used in the food industry (e.g. brands such as Aldi, REWE); household goods (e.g. IKEA) and textile industry (e.g. C&A).

regions. 3D models help to investigate geographical landscapes digitally and to develop a better understanding of existing resources. Employees and equipment can be captured and managed in the respective lifecycle phases. Even if almost all industries have a lifecycle, the composition of lifecycle phases differs significant. Therefore, this cannot be called product lifecycle management (PLM) anymore, but rather of a resource lifecycle management (RLM). A more broader and generic definition would avoid the explanation of a partial area and would speak about xLM (x corresponds to a variety of disciplines such as it is for CAx).

C.2 PDM Systems - People and Organisation

The listed user roles in Table C.1 are not identical to the user roles defined in PDM systems but give an insight into the hierarchical structure. In addition, individual user roles can be created and appropriate permissions can be assigned. The access permissions can be set for each user role separately. The authorization group typically includes permissions for *Read*, *Modify*, *Delete*, *CheckOut*, *CheckIn*, *Lock*, *Unlock*, *Execute*, *Freeze*, *Create*, *Revise*, *Promote*, *Demote*, *Grant*, *Revoke*, *Enable*, *Disable*, and *Override*.

C.3 PDM Dictionary

Table C.2 contains a summary of the ubiquitous term that is used in the PDM domain. A complete list of product-specific PLM terms can be found in the product documentation of the respective manufacturer.

Keyword	Definition
Business Object	Business objects in PDM systems represent objects of the business domain and include data, have relations to other objects, have an identity, and represents a primary component in the execution of business activities.

Continued on next page

Table C.2 – Continued from previous page

Keyword	Definition
Workspace	A data space consists of individuals, folders, content, tasks, meetings, access permissions to meet business requirements.
Folder	An object of the workspace that represents and summarizes other objects in a central location through a logical link. Folders are created for specific topics.
Process	An instance of a workflow that processes data. Typically, operations are performed with business objects, e.g. object state changes, revision generation, and creation of new relationships.
Property	Defines a characteristic state or value of a business object.
Relation	A persistent connection between objects.
Item / Type	A category in which business objects are assigned in order to distinguish them.
Trigger / Action Handler	An action which is performed when a specified event occur.
Attribute	A characteristic of a business object. Each field of a form references to an attribute such as description or weight of a business object.
BOM	Bill of Materials is a tabular and structured list of objects (e.g. parts and documents) of a more comprehensive object (e.g. assembly).
ECR	Engineering Change Request specifies the type of modification to a product or part.
ECO	Engineering Change Order specifies the activities of the modification based on the requirements of an ECR to a product or part.
Form	A compiled visual presentation of properties of a business object.

Continued on next page

Table C.2 – Continued from previous page

Keyword	Definition
Revision	A change to an original business object in which all features are frozen. A business object can be revised several times during the entire lifecycle.
Task	A sequence of individual steps in a workflow linked with other resources.
Document	A collection of files and (META) information.
Decision	An object which documents the decision based on the requirements.
Demote	Moves a business object from the current status to the previous state.
Promote	Moves a business object from the current state to the next state.
Policy	A set of rules for a business object type. A policy includes such as lifecycle states, access permissions, revision rules, etc.
Discussion	A conversation based on text-based messages with reference to a specific business object.
Drawing	A 2D-based design of views.

Table C.2: Ubiquitous terms of the PDM domain

User role	Description
Manager	Cross-project permissions. The role typically provide access to query reports and statistics in order to have an overview about the current project or program status.
Leader	Administrative rights for a team. They can add and remove team members to a project as well as manage tasks and user roles within the team. Typically, a team leader can see all documents of a team.
Designer	Creates content for the team. Content such as parts, assemblies, and documents are reviewed by other user roles. Once a document is submitted for review, it is no longer possible for the designer to do changes until the review process finished.
Viewer	Only rights to view content in read only mode. Document changes are not allowed. Viewers can usually examine assemblies only with a reduced informational content. Thus, not all data are available for the viewer.
Releaser	Releases content after a successful verification, so that business data assume a consistent and complete state
Reviewer	Performs reviews of documents, which were submitted by designers. Typically, such documents are not changed and only a review status and comment is added. Once the document review process is complete, the access permissions for the reviewers are revoked.

Table C.1: Generic user roles in the PDM system

D Mobile Feature Framework Modeling

D.1 Use Cases

The following use cases include mobile scenarios in PLM environments. The scenarios occur whenever a mobile user wants to achieve a business goal. The objective is achieved by the standard workflow and has previously met certain entry criteria. The result of individual steps in the standard workflow describes whether the execution was successful or not. A failure can at the same time be understood as a abort of the individual step or the entire scenario. The abort of the scenario means that the overall objective of the mobile user cannot be reached.

D.1.1 Discover Things by Shaking the Device

Group:	**Discover Objects**
Use Case:	**Discover Things by Shaking the Device**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User

Stakeholders and Interests:	Mobile Device User: Performs actions by shaking the mobile device.
Preconditions:	Mobile device is turned on, mobile device application is started, and mobile user is identified and authenticated.
Postconditions:	Defined action item is performed. Listening for the next shake event.

Main Success Scenario:

1. Mobile user shakes the mobile device
2. Mobile device application identifies the shake sequence
3. Mobile device application finds assigned action item for the shake sequence
4. Mobile device application executes defined action item

Extensions (or Alternative Flows):

2.a Kind of shake sequence not identifiable:
  1. Mobile device application shows failure message
  2. User returns to step 1 and corrects the errors

3.a Action item not assigned:
  1. Mobile device application shows failure message
  2. User returns to step 1

Special Requirements:	• Mobile device with attitude sensor • Mobile device with a mobile platform supporting mobile device applications • Multiple language support for displayed text.

Technology and Data Variations List:

2.a Different types of shake sequences

Frequency of Occurrence:	Nearly continuous in mobile situations

D.1.2 Discover Things by Gesture Recognizer

Group:	**Discover Objects**
Use Case:	**Discover Things by Gesture Recognizer**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User
Stakeholders and Interests:	Mobile Device User: Performs actions by gestures on the mobile device.
Preconditions:	Mobile device is turned on, mobile device application is started, and mobile user is identified and authenticated.
Postconditions:	Defined action item is performed. Listening for the next shake event.

Main Success Scenario:

1. Mobile user navigates to a web page using the mobile device application
2. Mobile user calls the 3D object on the web page
3. Mobile device application downloads the 3D object
4. Mobile device application opens the 3D object using a corresponding mobile feature
5. Mobile feature loads the explore mode
6. Mobile user explores the 3D object through gesture sequences
7. Mobile user terminates the explore mode

Extensions (or Alternative Flows):

2.a 3D object does not exist:
   1. Mobile device application shows failure message
   2. User returns to step 1

6.a Types of gesture sequences are not identifiable:
   1. Mobile device application shows failure message
   2. User returns to step 6 and corrects the errors

6.b Types of gesture sequences are not assigned:
   1. Mobile device application shows failure message
   2. User returns to step 6 and corrects the errors

Special Requirements:	• Mobile device with gesture recognition • Mobile device with a mobile platform supporting the mobile device applications • Multiple language support for displayed text.

Technology and Data Variations List:

6.a Various gesture recognizers

Frequency of Occurrence:	Nearly continuous in mobile situations

D.1.3 Discover Things by Speech Recognition

Group:	**Discover Objects**
Use Case:	**Discover Things by Speech Recognition**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User
Stakeholders and Interests:	Mobile Device User: Perform actions by speech on the mobile device.
Preconditions:	Mobile device is turned on, mobile device application is started, and mobile user is identified and authenticated.
Postconditions:	Defined action item is performed. Listening for the next shake event.

Main Success Scenario:

1. Mobile user shakes the mobile device
2. MDC identifies the shake sequence and calls the speech recognizer
3. Mobile user speaks a phrase
4. Speech recognizer identifies the language phrase sequence
5. Mobile application identifies assigned action element for identified language phrase sequence
6. Mobile application executes the defined action item

Extensions (or Alternative Flows):
2.a Types of gesture sequences are not assigned:
  1. Mobile device application shows failure message
  2. User returns to step 1 and corrects the errors
2.b Types of gesture sequences are not identifiable:
  1. Mobile device application shows failure message
  2. User returns to step 1 and corrects the errors
4.a Spoken word is not identified by speech recognition:
  1. Mobile device application shows failure message
  2. User returns to step 3
5.a Word is not assigned to an action item:
  1. Mobile device application shows failure message
  2. User returns to step 3	

Special Requirements:	• Mobile device with speech recognition • Mobile device with attitude sensor • Mobile device with a mobile platform supporting mobile device applications • Multiple language support for displayed text
Technology and Data Variations List: 2.a Various types of shake sequences 4.a Various types of speech sequences	
Frequency of Occurrence:	Nearly continuous in mobile situations

D.1.4 Discover Things by Facial Expression

Group:	**Discover Objects**
Use Case:	**Discover Things by Facial Expression**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User
Stakeholders and Interests:	Mobile Device User: Performs actions by facial expression on the mobile device.
Preconditions:	Mobile device is turned on, mobile device application is started, and mobile user is identified and authenticated.
Postconditions:	Defined action item is performed. Listening for the next facial expression event.

Main Success Scenario:

1. Mobile user starts the workflow
2. Workflow of the mobile application calls the facial expression recognizer
3. Mobile user makes a facial expression
4. Facial expression recognizer identifies the facial expression sequence
5. Mobile application identifies assigned action element for identified facial expression sequence
6. Execute defined action item

Extensions (or Alternative Flows):

1.a Missing access authorization for the workflow:
   1. Mobile device application shows failure message
   2. User returns to step 1

4.a Facial expression sequences is not identifiable:
   1. Mobile device application shows failure message
   2. User returns to step 3 and makes a different facial expression

5.a Types of facial expression sequences are not assigned:
   1. Mobile device application shows failure message
   2. User returns to step 3 and corrects the errors

Special Requirements:	• Mobile device with camera • Mobile device with facial expression recognition • Mobile device with a mobile platform supporting the mobile applications • Multiple language support for displayed text.

Technology and Data Variations List:

4.a Various types of facial expression

6.a Various types of individual mimic facets

Frequency of Occurrence:	Low application in mobile situations

D.1.5 Discover Things by User Context Consideration

Group:	**Discover Objects**
Use Case:	**Discover Things by User Context Consideration**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User
Stakeholders and Interests:	Mobile Device User: Improved consideration of user context in workflows and processes.
Preconditions:	Mobile device is turned on, mobile device application is started, and mobile user is identified and authenticated.
Postconditions:	User-specific context parameters are taken into account for the current activity of the mobile user.

Main Success Scenario:

1. Mobile user performs an activity
2. Mobile application requests user context parameter from mobile device application
3. Mobile device application determines the requested user context parameter
4. Mobile device application sends the user context parameter to the mobile application
5. Mobile application analyzes and processes the context parameter for the current activity
6. Mobile user perceives the context consideration for the current activity

Extensions (or Alternative Flows):

3.a User-specific context parameter is not defined:
1. Mobile application shows failure message
2. User returns to step 2 and corrects the errors

3.b User specific context parameters cannot be determined:
1. Mobile application shows failure message
2. User returns to step 2 and corrects the errors

2.a, 4.a Transmission of data is not possible:
1. Mobile device application shows failure message
2. User returns to the previous step and corrects the errors

6.a User specific context parameter cannot be taken into account for the activity:
1. Mobile application shows failure message
2. User returns to step 2 and corrects the errors

Special Requirements:	• Mobile device with sensors such as GPS sensor, motion sensor, audio, microphone, light sensor, temperature sensor • Mobile device with a mobile platform supporting mobile applications • Multiple language support for displayed text.

Technology and Data Variations List:

3.a Various data variations of the sensors

Frequency of Occurrence:	Nearly continuous in mobile situations

D.1.6 Observe User Behavior to Avoid Interruptions

Group:	**Discover Objects**
Use Case:	**Observe User Behavior to Avoid Interruptions**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User
Stakeholders and Interests:	Mobile Device User and Stationary User: Reduction of context interruptions.
Preconditions:	Mobile device is turned on, mobile device application is started, and mobile user is identified and authenticated.
Postconditions:	Mobile user will not disturb in the physical context and receives summary information about prevented context interruptions. The system learns new user behaviors and applies the acquired know-how in future context interruptions.

Main Success Scenario:

1. Mobile user configures the mobile device
2. Mobile user spends his attention to a physical context (e.g. meeting)
3. Mobile device application registers a physical context interrupt (e.g. receives an instant message)
4. Mobile device application collects information about the surrounding physical environment
5. Mobile device application decides whether the context interruption is approved against the device parameters
6. Mobile device application informs the mobile user about missed virtual contexts
7. Mobile device informs the source of interruption (e.g. Server or stationary user) about the mobile user state

---

Extensions (or Alternative Flows):

4.a Context parameters cannot be determined by the mobile device:
  1. Mobile device application permits context interruptions
  2. Mobile device application shows failure message
  3. Mobile user returns to step 1

5.a Context parameters are not sufficient:
  1. Mobile device application permits context interruptions
  2. Mobile device application shows failure message
  3. Mobile user returns to step 1

Special Requirements:	• Mobile device with GPS sensor, motion sensor, audio, microphone, light sensor, temperature sensor • Mobile device with a mobile platform supporting mobile applications • Multiple language support for displayed text.

Technology and Data Variations List:

4.a Different data variations from the sensors

Frequency of Occurrence:	Continuously in mobile situations

D.1.7 Identify Objects by Image Recognition

Group:	**Identify Objects**
Use Case:	**Identify Objects by Image Recognition**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User
Stakeholders and Interests:	Mobile Device User: Detection and identification of unknown real objects by image recognition of the mobile device.
Preconditions:	Mobile device is turned on, mobile device application is started and mobile user is identified and authenticated.
Postconditions:	Detected object is presented to the mobile user with object properties and object-related actions. Waiting for the next user action in order to recognize other objects through image recognition.

Main Success Scenario:

1. Mobile user calls the object recognition
2. Mobile user takes a photo of the object
3. Mobile device application identifies the objects in the captured photo
4. Mobile application links the identified object with the digital object
5. Mobile application provides the object results to the mobile user
6. Mobile user interprets returned information

Extensions (or Alternative Flows): 3.a Mobile device application cannot identify the object: 1. Mobile device application shows failure message 2. User returns to step 2 3.b Mobile device application identifies multiple objects: 1. Mobile device application lists all possible objects 4.a No information of the digital objects exists: 1. Mobile device application shows failure message 2. User returns to step 2 and corrects the errors	
Special Requirements:	• Mobile device with camera • Mobile device with a mobile platform supporting mobile applications • Multiple language support for displayed text.
Technology and Data Variations List: 2.a Various image recordings	
Frequency of Occurrence:	Nearly continuous in mobile situations

D.1.8 Identify Objects by ID-Code Scan of Objects

Group:	**Identify Objects**
Use Case:	**Identify Objects by ID-Code Scan of Objects**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User
Stakeholders and Interests:	Mobile Device User: Detection and identification of unknown real objects by ID-Code recognition through the mobile device.
Preconditions:	Mobile device is turned on, mobile device application is started, and mobile user is identified and authenticated.
Postconditions:	Detected object is presented to the mobile user with object properties and object-related actions. Waiting for the next user action to recognize other objects through ID-Code scan.

Main Success Scenario:

1. Mobile user calls the object recognition
2. Mobile user scans the ID code of the object
3. Mobile device application identifies the captured ID code of the object
4. Mobile application links the identified object with the digital object
5. Mobile application provides the object results to the mobile user
6. Mobile user interprets returned information

Extensions (or Alternative Flows):

3.a Mobile device application cannot identify the ID code of the object:
   1. Mobile application shows failure message
   2. User returns to step 2

4.a No information of the digital object exists:
   1. Mobile application shows failure message
   2. User returns to step 2 and corrects the errors

Special Requirements:	• Mobile device with camera • Mobile device with a mobile platform supporting mobile applications • Multiple language support for displayed text.

Technology and Data Variations List:

3.a Various types code schemas of the ID code (e.g. Barcode (EAN, UPC, IAN, JAN), Matrix-Codes (QR-Code, DataMatrix)

Frequency of Occurrence:	Nearly continuous in mobile situations

D.1.9 Identify Objects by Acoustics

Group:	**Identify Objects**
Use Case:	**Identify Objects by Acoustics**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User
Stakeholders and Interests:	Mobile Device User: Detection and identification of unknown real objects by acoustical recognition through the mobile device.
Preconditions:	Mobile device is turned on, mobile device application is started, and mobile user is identified and authenticated.
Postconditions:	Detected object is presented to the mobile user with object properties and object-related actions. Waiting for the next user action to recognize other objects through acoustical recognition.

Main Success Scenario:

1. Mobile user calls the object recognition
2. Mobile user captures acoustical noises of the object
3. Mobile device application identifies acoustical noises of the object through an acoustical fingerprint
4. Mobile application links the acoustical fingerprint of the real object with the digital object
5. Mobile application provides the object results to the mobile user
6. Mobile user interprets returned information

Extensions (or Alternative Flows):

- 3.a Mobile device application cannot identify the acoustical noises of the object:
  1. Mobile device application shows failure message
  2. User returns to step 2
- 3.b Mobile device application identify multiple possible objects:
  1. Mobile application lists all possible objects
- 4.a No information of the digital object exists:
  1. Mobile application shows failure message
  2. User returns to step 2 and corrects the errors

Special Requirements:	• Mobile device with microphone • Mobile device with a mobile platform supporting mobile applications • Multiple language support for displayed text.

Technology and Data Variations List:

- 3.a Various acoustical fingerprints of an object depending on the object state (e.g. Engine idling or engine with high rotation speed)

Frequency of Occurrence:	Low application in mobile situations

D.1.10 Explore Objects by Surfaces

Group:	**Explore Objects**
Use Case:	**Explore Objects by Surfaces**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User
Stakeholders and Interests:	Mobile Device User: Identification of objects by using the mobile device.
Preconditions:	Mobile device is turned on, mobile device application is started, and mobile user is identified and authenticated.
Postconditions:	The mobile user obtains the results of the object surface analysis and the identified object. Further object information can be retrieved. Waiting for the next user action to examine the surface of other objects.

Main Success Scenario:

1. Mobile user calls the object surface recognition
2. Mobile user chooses the measurement method
3. Mobile device application analyzes the measured data and identifies the respective object
4. Mobile application provides the object results to the mobile user
5. Mobile user interprets returned information

Extensions (or Alternative Flows):

2.a Measurement method does not exist:
1. Mobile device application shows failure message
2. User returns to step 2 in order to select other measurement method

3.a Measurement cannot be performed:
1. Mobile device application shows failure message
2. User returns to step 2

3.b Quality of the measurement data is insufficient:
1. Mobile device application shows failure message
2. User returns to step 2 in order to select other measurement method

3.c Object does not exist:
1. Mobile application shows failure message
2. User returns to step 2 in order to measure other objects

Special Requirements:	• Mobile device with camera • Mobile device with a mobile platform supporting mobile applications • Multiple language support for displayed text.

Technology and Data Variations List:

3.a Various captured data from the camera

Frequency of Occurrence:	Low application in mobile situations

D.1.11 Explore Objects by Measuring Characteristics

Group:	**Explore Objects**
Use Case:	**Explore Objects by measuring Characteristics**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User
Stakeholders and Interests:	Mobile Device User: Measurement of real object parameters using the mobile device
Preconditions:	Mobile device is turned on, mobile device application is started, and mobile user is identified and authenticated.
Postconditions:	Measurement results of the object or environment of the object are presented as well as object-related actions are provided to the mobile user. Waiting for the next user action to measure specific properties of other objects.

Main Success Scenario:

1. Mobile user calls the object recognition
2. Mobile user chooses the measurement method
3. Mobile device application performs the measurement
4. Mobile application recognizes the measurement results
5. Mobile application provides the results to the mobile user
6. Mobile user interprets returned information

Extensions (or Alternative Flows): 2.a Measurement method does not exist: 1. Mobile device application shows failure message 2. User returns to step 2 in order to choose another measurement method 3.a Measurement cannot be performed (e.g. outside of the value range): 1. Mobile device application shows failure message 2. User returns to step 2 or adjust measurement range 4.a Measurement data cannot be transited (e.g. network error): 1. Mobile device application shows failure message 2. User waits for data transmission 5.a No additional object properties exist: 1. Mobile device application shows message	
Special Requirements:	• Mobile device with sensors for thermometer and hydrometer, barometer, and light intensity • Mobile device with a mobile platform supporting mobile applications • Multiple language support for displayed text.
Technology and Data Variations List: 3.a Different data variations from the sensors 3.b Various measurement ranges	
Frequency of Occurrence:	Low application in mobile situations

D.1.12 Locate Objects by Coordinates

Group:	**Locate Objects**
Use Case:	**Locate Objects by Coordinates**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User
Stakeholders and Interests:	Mobile Device User: Locates objects and navigates to objects using the mobile device
Preconditions:	Mobile device is turned on, mobile device application is started, and mobile user is identified and authenticated.
Postconditions:	Additional object data and location information are presented to the mobile user when destination of the object has been reached. Object-related actions are provided to the mobile user. Waiting for the next user action to localize and navigate to other objects.

Main Success Scenario:

1. Mobile user calls an object
2. Mobile user chooses the object localization or navigation
3. Mobile device application determines the location information
4. Mobile device application navigates the mobile user to the object based on the location information
5. Mobile user follows the navigation instructions of the mobile device application

Extensions (or Alternative Flows):

3.a Location information does not exist for the object:
   1. Mobile application shows failure message
   2. User returns to step 1

4.a Navigation is not feasible because of incorrect location information:
   1. Mobile device application shows failure message
   2. User returns to step 1

4.b Navigation is not feasible because the object is located in-door:
   1. Mobile device application shows failure message
   2. User returns to step 1 in order to choose in-door navigation

5.a Mobile user does not follow the navigation instructions:
   1. Mobile device application shows message
   2. User returns to step 4

Special Requirements:	• Mobile device with GPS sensor • Mobile device with a mobile platform supporting mobile applications • Multiple language support for displayed text.

Technology and Data Variations List:

3.a Various data variations from the GPS sensor

Frequency of Occurrence:	Average application in mobile situations

D.1.13 Locate Objects by Acoustics

Group:	**Locate Objects**
Use Case:	**Locate Objects by Acoustics**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User
Stakeholders and Interests:	Mobile Device User: Localizes object types using the mobile device
Preconditions:	Mobile device is turned on, mobile device application is started, and mobile user is identified and authenticated.
Postconditions:	Mobile user can perform the object identification in order to obtain additional information and to execute object-related actions. Otherwise the mobile application will wait for the next user action to localize other object types through acoustics.

Main Success Scenario:

1. Mobile user calls the acoustic environment browser
2. Mobile user records the noises of object types
3. Mobile device application analyzes the recorded noise sequence
4. Mobile application searches the sequence in the database for related object types
5. Mobile user obtains the results visualized presented in the acoustic environment browser

Extensions (or Alternative Flows): 2.a Object types are noiseless: 1. Mobile device application shows failure message 2. User returns to step 2 3.a Recorded noise is too weak: 1. Mobile device application shows failure message 2. User returns to step 2 to improve the noise quality 4.a Noise sequence is not captured for this object type in the database: 1. Mobile device application shows failure message 2. User returns to step 2	
Special Requirements:	• Mobile device with microphone • Mobile device with a mobile platform supporting mobile applications • Multiple language support for displayed text.
Technology and Data Variations List: 3.a Different data variations of the audio sequences	
Frequency of Occurrence:	In mobile situations less continuously

D.1.14 Examine Objects by Augmented Reality

Group:	**Examine Objects**
Use Case:	**Examine Objects by Augmented Reality**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User
Stakeholders and Interests:	Mobile Device User: Investigation of physical objects with augmented reality using the mobile device.
Preconditions:	Mobile device is turned on, mobile device application is started, and mobile user is identified and authenticated.
Postconditions:	Mobile user receives additional information about the object environment and obtains object-related functionalities.

Main Success Scenario:

1. Mobile user calls the augmented reality
2. Mobile device application determines the location information
3. Mobile application determines the objects that are in the surrounding environment
4. Mobile device application presents these objects to the mobile user
5. Mobile user orients in the AR environment
6. Mobile user examines an individual real object
7. Mobile application determines additional object information

Extensions (or Alternative Flows):

2.a Location information cannot be determined:
  1. Mobile device application shows failure message
  2. User returns to step 2 and changes the location position

3.a No objects found at the location:
  1. Mobile device application shows failure message
  2. User returns to step 2 and changes the location position

7.a No additional object information available:
  1. Mobile device application shows failure message
  2. User returns to step 6 and examines another object

Special Requirements:	• Mobile device with GPS, compass, and motion sensors • Mobile device with a mobile platform supporting mobile applications • Multiple language support for displayed text.

Technology and Data Variations List:

2.a Different data variations from the sensors

Frequency of Occurrence:	Continuously in mobile situations

D.1.15 Examine Objects by Visualization

Group:	**Examine Objects**
Use Case:	**Examine Objects by Visualization**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User
Stakeholders and Interests:	Mobile Device User: Investigation of objects with visualization using the mobile device.
Preconditions:	Mobile device is turned on, mobile device application is started, and mobile user is identified and authenticated.
Postconditions:	Mobile user receives additional information about the object and obtains object-related functionalities.

Main Success Scenario:

1. Mobile user calls the visualization
2. Mobile device application determines the location information
3. Mobile application determines the objects that are in the surrounding environment
4. Mobile device application presents these objects to the mobile user
5. Mobile user orients in the visualization environment
6. Mobile user examines objects through visual filters
7. Mobile application determines additional object information

Extensions (or Alternative Flows):

2.a Location information cannot be determined:
  1. Mobile device application shows failure message
  2. User returns to step 2 and changes the location position

3.a No objects found at the location:
  1. Mobile application shows failure message
  2. User returns to step 2 and changes the location position

7.a No additional object information available:
  1. Mobile application shows failure message
  2. User returns to step 6 and examines another object

Special Requirements:	• Mobile device with GPS, compass, and motion sensors • Mobile device with a mobile platform supporting mobile applications • Multiple language support for displayed text.

Technology and Data Variations List:

2.a Various data variations from the sensors

Frequency of Occurrence:	Continuously in mobile situations

D.1.16 Overlay Realities to Simulate Part Assembly

Group:	**Examine Objects**
Use Case:	**Overlay Realities to Simulate Part Assembly**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User
Stakeholders and Interests:	Mobile Device User: Simplified assembly installation through Augmented Reality.
Preconditions:	Mobile device is turned on, mobile device application is started, and mobile user is identified and authenticated.
Postconditions:	Accelerated component installation leads to a faster start-up of the whole product. In addition, errors during assembly phase can be avoided without unclear descriptions in manuals. The user will see marked installation position on the real object provided by the visual assistance guide.

Main Success Scenario:

1. Mobile user calls the assembly wizard
2. Mobile application loads the required data
3. Mobile device application activates the camera and calls the augmented reality browser
4. AR browser identifies the real object
5. AR browser overlays the reality with additional visual information and visualized installation instructions
6. Mobile user follows the installation instructions
7. Mobile device application validates the overall condition of the composite component
8. Mobile device application transfers the component state information to the server
9. Server stores and connects the information to the maintenance service object and provides the result back to the mobile user
10. Mobile user interprets returned result

Extensions (or Alternative Flows):

2.a, 8.a Information cannot be transferred:
1. Mobile device application logs the faulty transfer
2. Mobile device application shows failure message
3. User returns to step 1 and corrects the errors

3.a Mobile device does not support augmented reality:
1. Mobile device application shows failure message
2. User returns to step 1 and corrects the errors

4.a Real object cannot be identified or is not supported:
1. Mobile device application shows failure message
2. User returns to step 1 and corrects the errors

6.a Mobile user does not follow the installation instructions:
1. Mobile device application shows failure message
2. User returns to step 5 and corrects the errors

7.a Component validation failed:
1. Mobile device application shows failure message
2. Mobile device application initiates the diagnosis and visualizes relevant diagnostic instructions
3. User returns to step 6 and corrects the errors

Special Requirements:	• Mobile device with GPS, compass, motion sensors, camera, and audio • Mobile device with a mobile platform supporting mobile applications • Multiple language support for displayed text. • Mobile photo editing software

Technology and Data Variations List:

4.a Different data variations from the sensors

Frequency of Occurrence:	Nearly continuous in mobile situations

D.1.17 Overlay Realities to Discover Facilities

Group:	**Examine Objects**
Use Case:	**Examine Objects to Discover Facilities**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User
Stakeholders and Interests:	Mobile Device User: Orientation and additional information in production facilities / Production Manager: Visualization and localization of production interventions
Preconditions:	Mobile device is turned on, mobile device application is started, and mobile user is identified and authenticated.
Postconditions:	The reality overlay with additional information of the production facility allows a faster overview of production steps in real time and response to problems. The user receives a summary overview about system power, system state and other production object parameters from the visual assistance, depends on the selected overlaid layer.

Main Success Scenario:

1. Mobile user calls the ProductionView-Wizard
2. Mobile application loads the required data from the server
3. Mobile device application activates the camera and calls the augmented reality module
4. AR module overlaps the reality with additional visual information
5. Mobile user selects an object from the production facility
6. Mobile device application loads the object details (e.g. composite components, object state)
7. Mobile application composes the results through an additional AR layer
8. Mobile device application visualizes the results for the mobile user
9. Mobile users interprets returned result

Extensions (or Alternative Flows):

2.a, 6.a Information cannot be transferred:
1. Mobile device application logs the faulty transfer
2. Mobile device application shows failure message
3. User returns to step 2 and corrects the errors

3.a Mobile device does not support augmented reality:
1. Mobile device application shows failure message
2. User returns to step 1 in order to choose other Wizard

4.a Real object cannot be identified or is not supported:
1. Mobile device application shows failure message
2. User moves the camera to another direction

Special Requirements:

- Mobile device with GPS, compass, motion sensors, camera, and audio
- Mobile device with a mobile platform supporting mobile applications
- Multiple language support for displayed text.
- Mobile photo editing software

Technology and Data Variations List:
2.a Various data variations of object types

Frequency of Occurrence:	Nearly continuous in mobile situations

D.1.18 Overlay Objects for BOM List Synchronization

Group:	**Examine Objects**
Use Case:	**Examine Objects for BOM List Synchronization**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User
Stakeholders and Interests:	Mobile Device User: Validation of BOM list contains real objects.
Preconditions:	Mobile device is turned on, mobile device application is started, and mobile user is identified and authenticated.
Postconditions:	The reality overlay with additional information leads to a comparison of real objects with items captured on a digital BOM list. Thereby, items on the BOM list that are not available as real object can be identified. Thus, inconsistencies such as missing and foreign real objects are identified immediately.

Main Success Scenario:

1. Mobile user calls the BOMView-Wizard
2. Mobile user chooses a BOM list
3. Mobile application loads the required data from the server
4. Mobile device application activates the camera and calls the augmented reality module
5. Mobile device application identifies real objects
6. Mobile device application performs a comparison between real objects and items on the BOM list
7. Mobile device application overlaps the reality with additional visual object information
8. Mobile user selects an object in the AR environment
9. Mobile device application loads object details
10. Mobile application composes the result through an extra AR layer
11. Mobile device application visualizes the result for the mobile user
12. Mobile user interprets returned result

Extensions (or Alternative Flows):

3.a, 9.a Information cannot be transferred:
  1. Mobile application logs the faulty transfer
  2. Mobile application shows failure message
  3. User returns to step 2 and corrects the errors

4.a Mobile device does not support augmented reality:
  1. Mobile device application shows failure message
  2. User returns to step 1 in order to call another Wizard

5.a Real object cannot be identified or is not supported:
  1. Mobile device application shows failure message
  2. User changes the position to the object

6.a Real objects cannot be matched against the BOM list:
  1. Mobile device application shows failure message
  2. User returns to step 2 and corrects the errors

Special Requirements:	• Mobile device with GPS, compass, motion sensors, camera, and audio • Mobile device with a mobile platform supporting mobile applications • Multiple language support for displayed text. • Mobile photo editing software
Technology and Data Variations List: 5.a, 6.a Various data sets	
Frequency of Occurrence:	Nearly continuous in mobile situations

D.1.19 Collaborate with Objects using Collaboration Tools

Group:	**Collaborate with Objects**
Use Case:	**Collaborate with Objects using Online Collaboration Tools**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User
Stakeholders and Interests:	Mobile Device User: Interactive communication enables rapid and better communication of information based on media content / Stationary User: Easier and faster understanding of the context through interactive content.
Preconditions:	Mobile device is turned on, mobile device application is started, and mobile user is identified and authenticated.
Postconditions:	Interactive content supports the mobile user to share facts from a mobile situation quickly and directly with other users. All users can influence the communication through interactive participation and thus move rapidly forward to a common result. Thereby, the result is required for subsequent process steps. Therefore, the stationary user obtains a better understanding of actually mobile contexts.

Main Success Scenario:

1. Mobile user calls the messaging service
2. Mobile user creates a group of participants and invites them
3. Mobile user chooses a virtual or real object
4. Mobile user chooses via a dialog the interactive method
5. Mobile device application loads the interactive component
6. Mobile users creates and modifies the object content interactively
7. Mobile device application transfers the content to the server
8. Server stores the content as copy of the original object and connects to the original object
9. Server provides the results back to the mobile user
10. Mobile user interprets returned information

Extensions (or Alternative Flows):

2.a Participants are not available:
   1. Mobile device application shows information message
   2. User invites other people

4.a Object does not support the interactive method:
   1. Mobile device application shows failure message
   2. User choose another interactive method

5.a, 7.a Data cannot be transferred:
   1. Mobile device application logs the faulty transfer
   2. Mobile device application shows failure message
   3. User returns waits until data transfer completed successfully or interrupt activity

Special Requirements:	• Mobile device with GPS, compass, camera, audio, and microphone • Mobile device with a mobile platform supporting mobile applications • Multiple language support for displayed text. • Mobile photo editing software

Technology and Data Variations List:

4.a Various formats for image (e.g. PNG, JPEG, GIF), video (e.g. MPEG, AVI), audio (e.g. MP3, WAV) as well as protocols for streaming (e.g. RTP, SIP)

Frequency of Occurrence:	In mobile situations less continuously

D.1.20 Remote Diagnostics and Device Control

Group:	**Collaborate with Objects**
Use Case:	**Remote Diagnostics and Device Control**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device
Stakeholders and Interests:	Mobile Device User: Assistance during problem analysis through remote diagnostics.
Preconditions:	Mobile device is turned on, mobile device application is started, and mobile user is identified and authenticated.
Postconditions:	The remote diagnosis ensures rapid and effective analysis of device-specific problems. The analysis result helps the user to perform action plans based on appropriate solutions developed by the system. Spare parts can be ordered at an early stage of the problem analysis.

Main Success Scenario:

1. Mobile user calls the Diagnose-Wizard
2. Mobile user follows the instructions of the Diagnostic-Wizards
3. Mobile user connects the mobile device with the faulty device or components on site
4. Mobile user starts the diagnosis and collects the data from the faulty device
5. Mobile device application transmits the diagnostic data to the server
6. Server evaluates the historical data and diagnostic data
7. Server proposes solutions and the action plan
8. Server transmits the result to the mobile device application
9. Mobile user interprets the returned result

Extensions (or Alternative Flows):

3.a Faulty device or component does not support diagnostics:
  1. Mobile device application shows failure message
  2. System returns to step 2

4.a Diagnostic data cannot be retrieved:
  1. Mobile device application records the diagnostic attempt
  2. System returns to step 2

5.a, 8.a System cannot transfer the data:
  1. Mobile device application logs the faulty transmission an starts a second attempt
  2. User returns to step 5

7.a Data evaluation by server is not possible (e.g. no historical data available):
  1. System creates an alternate action plan for the mobile user

---

Special Requirements:	• Mobile device with an interface (e.g. USB) for diagnostic devices or integrated sensors such as motion, audio, microphone, light intensity, temperature, hydrometer. In the second case the mobile device represents the diagnostic unit. • Mobile device with a mobile platform supporting mobile applications • Multiple language support for displayed text.

Technology and Data Variations List:
4.a Different data variations from the faulty device

Frequency of Occurrence:	In mobile situations less continuously

D.1.21 Authenticate Apps by Biometrical Characteristics

Group:	**Collaborate with Objects**
Use Case:	**Authenticate Application by Biometrical Characteristics**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User
Stakeholders and Interests:	Mobile Device User: Login to the PDM system using the mobile device
Preconditions:	-
Postconditions:	Mobile user obtains the user interface with all role-based functionalities after successful authentication.

Main Success Scenario:

1. Mobile device application invokes the authentication method
2. Mobile user authenticates with his face or fingerprint
3. Mobile application validates the authentication information
4. Mobile application shows the user interface with related functionalities

Extensions (or Alternative Flows):

2.a Mobile device does not support biometrical authentication:
  1. Mobile device application shows failure message
  2. User returns to step 1 in order to choose another authentication method

3.a Authentication of the mobile user is not possible:
  1. Mobile application shows failure message
  2. User returns to step 2 to try again or choose another authentication method

4.a Mobile user is not authorized:
  1. Mobile application shows failure message
  2. User returns to step 1

Special Requirements:	• Mobile device with facial recognition and fingerprint reader • Mobile device with a mobile platform supporting mobile applications • Multiple language support for displayed text.

Technology and Data Variations List:

2.a Different data variations of facial recognition and fingerprint reader

Frequency of Occurrence:	Continuously in mobile situations

D.1.22 Contribute Content by Media and Drawings

Group:	**Contribute Object Content**
Use Case:	**Contribute Content by self-recorded Media and Drawings**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User
Stakeholders and Interests:	Mobile Device User: Contribution of visual, verbal and auditory information / Stationary Users: Accelerated information recognition.
Preconditions:	Mobile device is turned on, mobile device application is started, and mobile user is identified and authenticated.
Postconditions:	Received multimedia must be attached to the specific object. Waiting for the next user action in order to capture additional multimedia content using the mobile device.

Main Success Scenario:

1. Mobile user opens the workflow interface
2. Mobile user selects a contribution method
3. Mobile device application loads the appropriate media component
4. Mobile user captures the multimedia content
5. Mobile device application transfers the multimedia content to the server
6. Server stores and connects the multimedia content to the related workflow object
7. Server transfers the corresponding result back to the mobile user
8. Mobile user interprets returned information

Extensions (or Alternative Flows):

2.a Workflow object does not support multimedia content:
  1. Mobile device application shows failure message
  2. User returns to step 1 in order to choose text-based contribution method

3.a Multimedia content type is not supported by the workflow object:
  1. Mobile device application shows failure message
  2. User returns to step 2 in order to choose multimedia contribution method

5.a, 7.a Multimedia data cannot be transfers (e.g. file size exceeded):
  1. Mobile device application shows failure message
  2. User returns to step 2 in order to choose multimedia contribution method

---

Special Requirements:	• Mobile device with camera and microphone • Mobile device with a mobile platform supporting mobile applications • Multiple language support for displayed text.

Technology and Data Variations List: 4.a Various formats for image (e.g. PNG, JPEG, GIF), video (e.g. MPEG, AVI) and audio (e.g. MP3, WAV)	
Frequency of Occurrence:	Nearly continuous in mobile situations

D.1.23 Contribute Content by Speech Recog. and Gestures

Group:	**Contribute Object Content**
Use Case:	**Contribute Content by Speech Recognition and Gestures**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User
Stakeholders and Interests:	Mobile Device User: Communication of decisions using voice recognition and gestures / Stationary Users: Accelerated communication of decisions.
Preconditions:	Mobile device is turned on, mobile device application is started, and mobile user is identified and authenticated.
Postconditions:	Communicated decisions will be used for subsequent process steps. Waiting for the next user action to communicate a decision using the mobile device.

Main Success Scenario:

1. Mobile user calls the workflow object
2. Mobile user makes a decision by gesture or voice
3. Mobile device application records the user decision
4. Mobile device application interprets the decision and creates a decision message
5. Mobile device application transfers the decision message to the server
6. Server stores and connects the decision message to the workflow object
7. Server executes the workflow-related action for the decision
8. Server provides the corresponding result back to the mobile user
9. Mobile user interprets returned result

Extensions (or Alternative Flows):

2.a Workflow does not support decision capturing by gesture or voice:
   1. Mobile device application shows failure message
   2. User returns to step 1 in order to perform a text-based decision

3.a Decision capturing type is not supported by the mobile device:
   1. Mobile device application shows failure message
   2. User returns to step 2 and corrects the errors

4.a Captured decision cannot be interpreted (e.g. bad pronunciation or surrounding noise):
   1. Mobile device application shows failure message
   2. User returns to step 3 and corrects the errors

5.a Decision message cannot be transferred to the server:
   1. Mobile device application logs the faulty transfer
   2. Mobile device application shows failure message
   3. User returns to step 5 and corrects the errors

Special Requirements:	• Mobile device with camera and microphone • Mobile device with a mobile platform supporting mobile applications • Multiple language support for displayed text.

Technology and Data Variations List:

2.a Various formats for image (e.g. PNG, JPEG, GIF), video (e.g. MPEG, AVI) and audio (e.g. MP3, WAV)

4.a Various gesture and voice interpretations

Frequency of Occurrence:	Nearly continuous in mobile situations

D.1.24 Contribute Content by filtering captured Objects

Group:	**Contribute Object Content**
Use Case:	**Contribute Content by filtering captured Objects**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device User
Stakeholders and Interests:	Mobile Device User: Simplifies image description and labeling of objects by the user / Stationary Users: Rapid understanding of the facts through pictures for further user activities.
Preconditions:	Mobile device is turned on, mobile device application is started, and mobile user is identified and authenticated.
Postconditions:	A faster understanding of facts by the mobile user leads to an accelerated processing of subsequent process steps. Moreover, errors are avoided which are caused by misunderstandings. The mobile user can move through marked positions of the image and can directly jump into the editing mode of the corresponding object.

Main Success Scenario:

1. Mobile user opens the workflow interface
2. Mobile user creates a picture of an object related to the workflow
3. Mobile device application captures the picture and switches to the image edit mode
4. Mobile user modifies the picture (e.g. Marks surfaces)
5. Mobile device application transfers the picture file to the server
6. Server stores and connects the picture file to the related object
7. Server provides the corresponding result back to the mobile user
8. Mobile user interprets returned result

Extensions (or Alternative Flows):

2.a Mobile device does not support picture capturing:
   1. Mobile device application shows failure message
   2. User returns to step 1 in order to capture text-based information

3.a Picture format is not supported:
   1. Mobile device application shows failure message
   2. User returns to step 2 and corrects the errors

4.a Captured picture cannot be modified (e.g. poor image quality):
   1. Mobile device application shows failure message
   2. User returns to step 2 and corrects the errors

5.a Image file cannot be transferred to the server:
   1. Mobile device application logs the faulty transfer
   2. Mobile device application shows failure message
   3. User returns to step 5 and corrects the errors

Special Requirements:	• Mobile device with camera • Mobile device with a mobile platform supporting mobile applications • Multiple language support for displayed text. • Mobile photo editing software

Technology and Data Variations List:

2.a Various formats for image (e.g. PNG, JPEG, GIF)

3.a Various mobile photo editing software

Frequency of Occurrence:	Nearly continuous in mobile situations

D.1.25 Autonomous Transmission of Measurement Data

Group:	**Autonomous Object Actions**
Use Case:	**Autonomous Transmission of Measurement Data**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device
Stakeholders and Interests:	Mobile Device User: Unattended operations of autonomous mobile devices / Stationary User: Unattended operations of autonomous mobile devices.
Preconditions:	Mobile device is turned on, mobile device application is started, identified, and authenticated.
Postconditions:	Processing of the transmitted data for further activities.

Main Success Scenario:

1. Mobile device is configured for autonomous data transfer and collection of measurements
2. Mobile device application authenticates against the server and starts the data collection
3. Mobile device application collects information of the physical environment
4. Mobile device application transfers collected data to the server
5. Mobile device application receives new configuration data for self-control and optimization
6. Mobile device application informs the server about the mobile device state

Extensions (or Alternative Flows):

3.a Mobile device cannot collect data of the physical environment:
  1. Mobile device application informs the server
  2. Server performs a self-diagnostic and tries to correct the error
  3. Server returns to step 3

4.a Data transfer to the server failed:
  1. Mobile device application logs the faulty transmission
  2. System returns to step 4

5.a Configuration data invalid:
  1. Mobile device application requests new configuration data
  3. User returns to step 5

Special Requirements:	• Mobile device with sensors such as motion, audio, microphone, light intensity, temperature, hydrometer • Mobile device with a mobile platform supporting mobile applications • Multiple language support for displayed text.

Technology and Data Variations List:

3.a Various data variations from the sensors

Frequency of Occurrence:	Continuously in mobile situations

D.1.26 Mobile Device Calling Home

Group:	**Autonomous Object Actions**
Use Case:	**Mobile Device Calling Home**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device
Stakeholders and Interests:	Mobile Device: Error prevention through vitality data communication
Preconditions:	Mobile device is turned on, mobile device application is started, identified and authenticated (commissioning). The data connection requires a mobile network to transmit the vitality state of the mobile device to the server.
Postconditions:	The transmitted data are analyzed and archived.

Main Success Scenario:

1. Mobile device application transmits regularly heartbeats over the mobile network to the server
2. Mobile device application receives confirmation of transmission
3. Mobile device application interprets the confirmation and if necessary initiates further adjustment

Extensions (or Alternative Flows):

1.a, 2.a Data transfer not possible:

1. Mobile device application records the data transfer and starts a new attempt
2. Mobile device application returns to step 1

Special Requirements:	• Mobile device with access to a mobile network. • Mobile device with a mobile platform supporting mobile applications. • Multiple language support for displayed text.
Technology and Data Variations List: 1.a Various communication protocols and formats	
Frequency of Occurrence:	Continuously in mobile situations

D.1.27 Communication to Adapt the Behavior

Group:	**Autonomous Object Actions**
Use Case:	**Communication to Adapt the Behavior**
Scope:	Mobile Device
Level:	User-goal
Primary Actor:	Mobile Device
Stakeholders and Interests:	• Mobile Device: Supports the communication to other devices in order to optimize behavior changes.
Preconditions:	Mobile device is turned on, mobile device application is started, identified, and authenticated (commissioning). From the mobile device perspective, the communication has to be ensured in order to change device's behavior in conjunction with other devices. The communication link is required to exchange information.
Postconditions:	The information analysis represents a downstream task of the server in order to avoid contradictory behavior.

Main Success Scenario:

1. Mobile device application requests behavior information over the network
2. Mobile device application changes its own behavior based on the overall composite behavior of other devices
3. Mobile device application regularly receives new behavior changes from other devices
4. Mobile device application decides whether behavioral changes of the device group requires its own behavioral changes
5. Mobile device application changes its own behavior
6. Mobile device application communicates its own behavioral adaptation to other devices
7. Other devices confirm the behavioral adaptation
8. Mobile device application interprets the confirmation and if necessary, initiates further adjustment

---

Extensions (or Alternative Flows):

1.a, 3.a, 6-7.a Data transfer not possible:
   1. Mobile device application records the data transfer and starts a new attempt
   2. System returns to the previous step

4.a Decision to change the own device behavior can not be determined based on the retrieved data from other devices:
   1. Mobile device application communicates this information to the composite device group
   2. System returns to step 3

5.a Own behavior cannot be adjusted as expected, because it is out of the tolerance:
   1. Mobile device application changes the behavior to the maximum limit of tolerance
   2. User returns to step 3

Special Requirements:	• Mobile device with an interface (e.g. USB) to target devices or integrated sensors such as motion, audio, microphone, light intensity, temperature, hydrometer. In the second case the mobile device represents the target device. • Mobile device with a mobile platform supporting mobile applications • Multiple language support for displayed text.
Technology and Data Variations List: 4.a Different data variations from the sensors	
Frequency of Occurrence:	In mobile situations less continuously

D.2 Business Activities

This section describes a number of feature lists with some individual features which are described according to the scheme *‹Action›‹Result›‹Object›*. Therefore, all involved components of the framework are listed for each feature separately.

Id	*‹Action›‹Result›‹Object›*	Component(s)
SP01	Create voice command list for mobile users	MDF
SP02	Observe Shake-pattern of the mobile device	MDC, MDF
SP03	Transform language to words on the mobile device	MDC
SP04	Download voice command list from the server	MDC, MDF
SP05	Compare suggested words with command list on the mobile device	MDC
SP06	Perform the activity on the mobile device	MDC, MDF
SP07	Show results to the mobile user	MDC

Table D.28: Business activities: Speech Recognizer

Id	*‹Action›‹Result›‹Object›*	Component(s)
OI01	Start barcode scanner on mobile device	MDC
OI02	Activate the camera of the mobile device	MDC
OI03	Inform the user to position the camera on the ID code	MDC
OI04	Create a photo of the ID code	MDC
OI05	Determine the ID code of the object	MDC
OI06	Transmit the ID code to the web-based application (MDA)	MDC, MDF, MDA
OI07	Query the object with the associated ID code in the PDM system	MDA, MDF, MDI
OI08	Show the results of the PDM query to the mobile user	MDC, MDF, MDA

Table D.29: Business activities: Object ID Identifier

Id	*‹Action›‹Result›‹Object›*	Component(s)
OR01	Activate the camera of the mobile device	MDC
OR02	Inform the user to position the camera to the object	MDC
OR03	Create a photo of the object	MDC
OR04	Transfer the photo of the object to image recognition service	MDC
OR05	Determine the Id of the detected object	MDC
OR06	Send the Id to web-based application (MDA)	MDC, MDF, MDA
OR07	Query the object with the associated ID code in the PDM system	MDA, MDF, MDI
OR08	Show the results of the PDM query to the mobile user	MDC, MDF, MDA

Table D.30: Business activities: Object Recognition Identifier

Id	*‹Action›‹Result›‹Object›*	Component(s)
AB01	Download the config. for the Data Collector	MDC, MDF
AB02	Collect data on the mobile device	MDC
AB03	Authenticate data collector to framework	MDC, MDF
AB04	Transfer data to the framework	MDC, MDF
AB05	Store data on a queue	MDF
AB06	Inform framework about mobile device status	MDC, MDF
AB07	Load the data from the queue	MDF, MDA
AB08	Analyze collected data from mobile device	MDA

Table D.31: Business activities: Autonomous BOT

Id	*‹Action›‹Result›‹Object›*	Component(s)
MC01	Activate the camera of the mobile device	MDC
MC02	Activate the microphone of mobile device	MDC
MC03	Download the configuration options to be created for the media content	MDC, MDF, MDA
MC04	Create a photo of the object	MDC
MC05	Create a video of the object	MDC
MC06	Capture the voice/noise of the object	MDC
MC07	Edit the recorded photo of the mobile user	MDC
MC08	Edit the recorded video of the mobile user	MDC
MC09	Transfer the file to the application (MDA)	MDC, MDF, MDA
MC10	Save the file to the database	MDF, MDI, MDA
MC11	Show results to the mobile user	MDC, MDF, MDA

Table D.32: Business activities: Media Contributor

Id	*‹Action›‹Result›‹Object›*	Component(s)
OP01	Determine config. options for the feature	MDF, MDA
OP02	Determine the object data for the feature	MDF, MDA, MDI
OP03	Collect the config. option for the feature	MDA
OP04	Collect the object data for the feature	MDA
OP05	Transfer the configuration options and object data to the feature	MDA, MDF, MDC
OP06	Validate config. options and object data for the feature	MDC
OP07	Load config. options and object data for the feature	MDC
OP08	Determine the location of the mobile user	MDC
OP09	Start the feature on the mobile device	MDC
OP10	Show visualized objects to the mobile user	MDC

Table D.33: Business activities: Environmental Object Presenter

Id	*‹Action›‹Result›‹Object›*	Component(s)
SC01	Load the config. options for feature	MDC, MDF
SC02	Recognize vibrations of the mobile device	MDC
SC03	Activate microphone of the mobile device	MDC
SC04	Inform mobile user to provide a keyword	MDC
SC05	Identify name and role of the mobile user	MDC, MDF, MDI
SC06	Download keyword list of the mobile user	MDC, MDF
SC07	Compare speech recognition results with keywords list	MDC
SC08	Start activity for the defined keyword	MDC, MDA
SC09	Show the activity to the mobile user	MDC

Table D.34: Business activities: Speech Commander

Id	*‹Action›‹Result›‹Object›*	Component(s)
LE01	Send location request to the mobile device	MDA, MDF, MDC
LE02	Determine the location of the mobile device	MDC
LE03	Transfer location data to MDA application	MDC, MDF, MDA
LE04	Store the location data in the user session	MDA
LE05	Show location info to mobile user	MDA

Table D.35: Business activities: User Location Examinator

Id	*‹Action›‹Result›‹Object›*	Component(s)
FA01	Start the feature on the mobile device	MDC, MDF, MDA
FA02	Read the fingerprint of the mobile user	MDC
FA03	Transfer and validate the fingerprint string of the mobile user on the server	MDC, MDF, MDA
FA04	Determine the user roles for the mobile user	MDF, MDA, MDI
FA05	Transmit the validation result to the mobile device	MDF, MDA, MDC
FA06	Set the context of the application for the mobile user according to the user roles	MDF, MDA, MDI
FA07	Check if the mobile device supports the verification by fingerprint	MDF, MDA, MDC
FA08	Check whether a fingerprint on the mobile device is already registered	MDF, MDA, MDC
FA09	If the fingerprint check fails, provide the option to enter the user password	MDF, MDA, MDC
FA10	Transfer the password of the mobile user to the server and validate the password	MDF, MDA, MDC, MDI
FA11	Set maximum validity time for successful authentication (access conditions)	MDF, MDA, MDI

Table D.36: Business activities: Feature Authenticator

Id	*‹Action›‹Result›‹Object›*	Component(s)
UI01	Send context request to the mobile device	MDA, MDF, MDC
UI02	Determine mobile device context data	MDC
UI03	Transfer the context data to the application (MDA)	MDC, MDF, MDA
UI04	Store the context data in the user session	MDA
UI05	Show the context info to mobile user	MDA

Table D.37: Business activities: User Interrupter

Id	*‹Action›‹Result›‹Object›*	Component(s)
EC01	Determine context info of mobile device	MDC
EC02	Transfer context info to application (MDA)	MDC, MDF, MDA
EC03	Store context info in user session	MDA
EC04	Read volume level from context info	MDA
EC05	Read config options for volume level from the database	MDC, MDF
EC06	Compare config options with volume level info provided by the context	MDA
EC07	Change the volume level of the mobile device based on the specified volume level information	MDC, MDF, MDA
EC08	Show the modified volume level to the mobile user	MDA, MDF, MDC

Table D.38: Business activities: User Environment Controller

Id	*‹Action›‹Result›‹Object›*	Component(s)
BE01	Determine the context information of the mobile device	MDC
BE02	Transfer the context information to application (MDA)	MDC, MDF, MDA
BE03	Store context info in user session	MDA
BE04	Read brightness from context info	MDA
BE05	Read the configuration options for brightness from the database	MDC, MDF
BE06	Compare config. options with brightness info provided by context info	MDA
BE07	Change application styling based on the specified brightness info	MDA
BE08	Show the modified UI to the mobile user	MDA

Table D.39: Business activities: Brightness Examinator

Id	*‹Action›‹Result›‹Object›*	Component(s)
OA01	Determine config options for the feature	MDF, MDA
OA02	Determine the object data for the feature	MDF, MDA, MDI
OA03	Collect config option for the feature	MDA
OA04	Collect the object data for the feature	MDA
OA05	Transfer the configuration options and object data to the feature	MDA, MDF, MDC
OA06	Validate config options and object data	MDC
OA07	Load the config options and object data	MDC
OA08	Determine the location of the mobile user	MDC
OA09	Recognize the real object on site	MDC
OA10	Start the feature on the mobile device	MDC
OA11	Show visualized objects to the mobile user	MDC
OA12	Show assembly instruction to mobile user	MDC
OA13	Load object data into augmented reality	MDC

Table D.40: Business activities: Object Assembler

D.3 Services, Repositories, and Events

Comp.	Layer	Service Name	Task
MDC	AL	MobileDevice ClientService	Configure device client, Obtain presentation model
MDC	DL	MobileFeature ClientService	Process incoming mobile feature request and deliver the results
MDC	DL	Message Protocol	Usage of specific message protocols
MDC	IL	Transport Service	Establish connections to MDF channel
MDC	IL	Messaging Service	Receive and send messages from/to MDF

Table D.41: Services in the MDC component

Comp.	Layer	Service Name	Task
MDF	AL	MobileDevice Service	Processing of device specific tasks
MDF	AL	MobileApplication Service	Processing of application specific tasks
MDF	DL	MobileFeature Service	Validation and Processing of feature requests
MDF	DL	NotificationService	Processing of notification to MDC and MDA
MDF	DL	BusinessObject-HandlerService	Request processing to MDI and delivery of Business Object Information to MDA
MDF	DL	MessageProtocol	Usage of message protocols
MDF	IL	MessagingService	Receive and send messages

Table D.42: Services in the MDF component

Comp.	Layer	Service Name	Task
MDI	AL	Integration-Service	Request processing and delivery of Business Object Information to MDF
MDI	DL	Integration	Process Business Object Requests and validate permissions
MDI	DL	Business-ObjectService	Delegate task for reading, updating, storing to Infrastructure Layer
MDI	DL	Notifier	Processing of notification from PDM system to MDF
MDI	IL	BusinessObject LegacyService	Mapping Generic Business Object from Domain Layer to Legacy Business Object from Infrastructure Layer/PDM system
MDI	IL	Relationship LegacyService	Processing relationship related tasks of the legacy system
MDI	IL	Transport Service	Establish a connection channel between MDI and PDM system (e.g. Webservice/JSON)
MDI	IL	Messaging-Service	Receive and Send messages between MDI and PDM system

Table D.43: Services in the MDI component

Comp.	Layer	Service Name	Task
MDA	AL	Application Service	Processing UI action, Obtain presentation model
MDA	DL	FeatureService	Processing mobile feature events
MDA	DL	BusinessService	Processing business object events

Table D.44: Services in the MDA component

Comp.	Layer	Repository Name	Task
MDC	IL	EventStoreRepository	Storage for received and delivered messages
MDC	IL	MessageProtocolImpl	Configuration options for message delivery
MDC	IL	MobileFeaturePluginImpl	Storage for feature list configuration

Table D.45: Repositories in the MDC component

Comp.	Layer	Repository Name	Task
MDA	IL	ApplicationEventStore Repository	Storage of received and delivered messages
MDA	IL	ViewRepositoryImpl	Storage for the Views and View configuration

Table D.46: Repositories in the MDA component

Comp.	Layer	Repository Name	Task
MDI	IL	Relationship-LegacyImpl	Mapping relationship operations between business objects
MDI	IL	BusinessObject-LegacyImpl	Mapping business object operations

Table D.47: Repositories in the MDI component

Comp.	Layer	Repository Name	Task
MDF	IL	MobileDeviceImpl	Storage for device specific configurations and runtime values
MDF	IL	Communication-ChannelImpl	Storage for channel definitions of applications
MDF	IL	EventStore-Impl	Storage for incoming feature messages and outgoing events send to mobile applications
MDF	IL	MobileFeature-Impl	Storage for feature configuration options (e.g. Take three pictures in sequence)
MDF	IL	Mobile-ApplicationImpl	Storage for application information (e.g. used communication channels and availability)
MDF	IL	Message-ProtocolImpl	Storage for Configuration options for message delivery

Table D.48: Repositories in the MDF component

Comp.	Layer	Event Name	Task
MDC	DL	Feature RequestEvent	Triggered by a mobile feature request
MDC	DL	Feature ResponseEvent	Triggered as a result of a mobile feature request
MDC	DL	Notification Event	Notifications triggered by MessagingService (Failed message delivery), TransportService (connection issues), and MDF for object-based notifications
MDC	IL	DataEvent	Triggered by mobile feature to transmit continuous data streams to MDA

Table D.49: Events in the MDC component

Comp.	Layer	Event Name	Task
MDA	DL	MobileFeature Request	Trigger a mobile feature request
MDA	DL	MobileFeature Response	Event is triggered as a result of a mobile feature request
MDA	DL	BusinessObject RequestEvent	Event is triggered for a business object request
MDA	DL	BusinessObject ResponseEvent	Event is triggered as a result of a business object request
MDA	DL	Application NotificationEvent	Present notifications triggered by MDI
MDA	IL	DataEvent	Triggered by MDC mobile feature to transmit continuous data streams to MDA

Table D.50: Events in the MDA component

Comp.	Layer	Event Name	Task
MDI	DL	Notification-Event	Triggered by MDI to deliver generic notifications to MDF
MDI	DL	Legacy-NotificationEvent	Triggered by the PDM system to deliver legacy notifications (e.g. Subscription to Business operations like Add/Remove content)
MDI	DL	BusinessObject-RequestEvent	Triggered by MDF and used for data processing in the PDM system (Create BO, Save Files)
MDI	DL	BusinessObject-ResponseEvent	Triggered as a response for a business object request to deliver data to MDF

Table D.51: Events in the MDI component

Comp.	Layer	Event Name	Task
MDF	DL	MobileFeature RequestEvent	Event is received as a mobile feature request from MDA
MDF	DL	MobileFeature Respon-seEvent	Event is created as response to the mobile feature request which is used to triggered the message delivery from MDC to MDA
MDF	DL	Notification Event	Event is triggered by MDI, received by MDF, and is used to triggered the message delivery to the MDC
MDF	DL	BusinessObject RequestEvent	This event is used to request business object information
MDF	DL	BusinessObject Respon-seEvent	Event is triggered by MDI and used for data delivery to MDA for MDA data processing as well as save file content in the PDM system via MDI which was captured by MDC.
MDF	DL	DataEvent	This event is used for continuous data streams from MDC to MDA

Table D.52: Events in the MDF component

Bibliography

[3GP09] 3GPP. International Mobile station Equipment Identities (IMEI). http://www.3gpp.org/ftp/Specs/archive/22_series/22.016/22016-900.zip, 2009. [Retrieved: 13.03.2013].

[Abr09] Michael Abramovici. Benefits of PLM in der Automobilindustrie: Ergebnisse einer Benchmark-Studie. http://www.mesago.de/v3/download.php?name=Abramovici2.pdf&c_id=8000&file=00903066001268217178.pdf, 2009. [Retrieved: 07.01.2013].

[Acc10] Accenture. Product Lifecycle Management - Innovation umsetzen. http://www.accenture.com/SiteCollectionDocuments/Local_Germany/PDF/Accenture_Broeschuere_PLM_Innovationen_umsetzen_2011.pdf, 2010. [Retrieved: 07.01.2013].

[ACL07] ACL Engineering Ltd. Planned Maintenance. http://www.aclengineering.co.uk/plannedmaintenance.htm, 2007. [Retrieved: 12.11.2014].

[Ada10] David Adams. How Financial Services can use Product Lifecycle Management to ensure regulatory compliance. http://www.hitachiconsulting.com/files/pdfRepository/UK_Product_Lifecycle_Management_White_Paper_February_2010_FINAL.pdf, 2010. [Retrieved: 31.08.2013].

[AGM+10] Adam J. Aviv, Katherine Gibson, Evan Mossop, Matt Blaze, and Jonathan M. Smith. Smudge Attacks on Smartphone Touch Screens. In *Proceedings of the 4th USENIX Conference on Offensive Technologies*, WOOT'10, pages 1–7, Berkeley, CA, USA, 2010. USENIX Association.

[AK11] Iosif Androulidakis and Gorazd Kandus. A Survey on Saving Personal Data in the Mobile Phone. In *2011 Sixth International Conference on Availability, Reliability and Security*, pages 633–638. IEEE, 2011.

[AK12] Iosif Androulidakis and Gorazd Kandus. Feeling Secure vs. Being Secure the Mobile Phone User Case. In Christos K. Georgiadis, Hamid Jahankhani, Elias Pimenidis, Rabih Bashroush, and Ameer Al-Nemrat, editors, *Global Security, Safety and Sustainability & e-Democracy*, volume 99 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 212–219. Springer Berlin Heidelberg, 2012.

[Amb12] Scott W. Ambler. Feature Driven Development (FDD) and Agile Modeling. `http://www.agilemodeling.com/essays/fdd.htm`, 2012. [Retrieved: 12.01.2014].

[Ang06] Roland Angert. *Praktische Maschinenakustik*. VDI-Buch. Springer, Berlin, 2006.

[Ash09] Kevin Ashton. That 'Internet of Things' Thing. `http://www.rfidjournal.com/articles/view?4986`, 2009. [Retrieved: 15.12.2013].

[Ass12] TYPO3 Association. ECT - TYPO3Wiki - MVC Framework for Extensions. `http://wiki.typo3.org/ECT{#}MVC_Framework_for_Extensions`, 2012. [Retrieved: 17.09.2013].

[Bal07] Barbara Ballard. *Designing the mobile user experience*. Wiley, Chichester, 2007.

[Bal09] Helmut Balzert. *Lehrbuch der Software-Technik. / [1], Basiskonzepte und requirements engineering*. Lehrbücher der

Informatik. Spektrum, Akad. Verl., Heidelberg [u.a.], 3rd edition, 2009.

[Bar08] Barry Collins. Mobile phone projectors "will launch this year". http://www.pcpro.co.uk/news/153237/mobile-phone-projectors-will-launch-this-year, 2008. [Retrieved: 20.12.2012].

[BB91] John M. Bowers and Steve Benford. *Studies in Computer Supported Cooperative Work: Theory, Practice, and Design (Human Factors in Information Technology)*, volume 8 of *Human Factors in Information Technology*. North-Holland, Amsterdam; Oxford, 1991.

[BBC00] John Bransford, Ann L. Brown, and Rodney R. Cocking. *How People Learn: Brain, Mind, Experience, and School*. National Academy Press, Washington, D.C, expanded edition, 2000.

[BGT05] Dirk Balfanz, Matthias Grimm, and Mohammad-Reza Tazari. A Reference Architecture for Mobile Knowledge Management. In Nigel Davies, Thomas Kirste, and Heidrun Schumann, editors, *Mobile Computing and Ambient Intelligence: The Challenge of Multimedia*, Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2005. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.

[BH98] Hugh Beyer and Karen Holtzblatt. *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann, San Francisco, Calif, 1998.

[Bis03] Matt Bishop. *Computer Security: Art and Science*. Addison-Wesley, Boston, 2003.

[Bon12] Joseph Bonneau. The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords. In *2012 IEEE Symposium on Security and Privacy*, pages 538–552. IEEE, 2012.

[Bou11] Jamila Boughelaf. Mobile phones, social media and the Arab Spring. http://credemus.org/images/stories/repo rts/mobile-phones-and-the-arab-spring.pdf, 2011. [Retrieved: 14.05.2013].

[BSM12] Jan Lauren Boyles, Aaron Smith, and Mary Madden. Privacy and Data Management on Mobile Devices: More than half of app users have uninstalled or avoided an app due to concerns about personal information. http://pewinternet.org/~/media//Files/Reports/2012/PIP_MobilePrivacyManagement.pdf, 2012. [Retrieved: 07.03.2013].

[Bul08] Rebecca Bulander. *Customer-Relationship-Management-Systeme unter Nutzung mobiler Endgeräte*. Universitätsverlag, Karlsruhe, 2008.

[Bun12] Bundesamt für Sicherheit in der Informationstechnik. Leitfaden Informationssicherheit: IT-Grundschutz kompakt. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Leitfaden/GS-Leitfaden_pdf.pdf?__blob=publicationFile, 2012. [Retrieved: 13.03.2013].

[Bus00] Frank Buschmann. *Pattern-orientierte Software-Architektur: Ein Pattern-System*. Professionelle Softwareentwicklung. Addison-Wesley, Bonn [u.a.], 1st edition, 2000.

[BvB04] Marc Baumgart and Jan van Bon. *IT-Service-Management: Eine Einführung; basierend auf ITIL*. Van Haren, [Zaltbommel], 2nd edition, 2004.

[Can12] Naomi Canton. Cell phone culture: How cultural differences affect mobile use. http://edition.cnn.com/2012/09/27/tech/mobile-culture-usage, 2012. [Retrieved: 13.05.2013].

[CB12] Brian X. Chen and Nick Bilton. Et Tu, Google? Android Apps Can Also Secretly Copy Photos. http://bits.blogs.nytimes.com/2012/03/01/android-photos/, 2012. [Retrieved: 08.05.2013].

[CD13] Pamela Clark-Dickson. Press release: OTT messaging traffic will be twice the volume of P2P SMS traffic by end-2013: The volume of OTT messaging traffic is set to be twice that of P2P SMS messaging by the end of the year, according to data collected by Informa Telecoms & Media. http://blogs.informatandm.com/12861/news-release-ott-messaging-traffic-will-be-twice-the-volume-of-p2p-sms-traffic-by-end-2013/, 2013. [Retrieved: 06.06.2013].

[CDV08] David Chen, Guy Doumeingts, and François Vernadat. Architectures for enterprise integration and interoperability: Past, present and future. *Enterprise Integration and Interoperability in Manufacturing Systems*, 59(7):647–659, 2008.

[CE05] John Clarkson and Claudia Eckert. *Design Process Improvement: A review of current practice*. Springer, London [U.K.], 2005.

[Chr12] Stefan Christmann. *Mobiles Internet im Unternehmenskontext: Webtechnologien als technische Basis für Geschäftsanwendungen auf mobilen Endgeräten*. Universitätsverlag Göttingen, [s.l.], 2012.

[Chu08] Eric Chu. Android Market: Now available for users. http://android-developers.blogspot.de/2008/10/android-market-now-available-for-users.html, 2008. [Retrieved: 20.12.2012].

[Chu12] Wai-chi Rodney Chu. *Mobile Communication and Greater China*, volume 1 of *Routledge Research on Social Work, Social Policy and Social Development in Greater China*. Routledge, Milton Park, Abingdon, Oxon, New York, 2012.

[CJP09] Jaime Campos, Erkki Jantunen, and Om Prakash. A web and mobile device architecture for mobile e-maintenance. *The International Journal of Advanced Manufacturing Technology*, 45(1):71–80, 2009.

[Cow11] James Cowie. Egypt Leaves the Internet. http://www.rene sys.com/blog/2011/01/egypt-leaves-the-internet.shtml, 2011. [Retrieved: 14.05.2013].

[DA00] Anind K. Dey and Gregory D. Abowd. Towards a Better Understanding of Context and Context-Awareness. In *Computer Human Intraction 2000 Workshop on the What, Who, Where, When, Why and How of Context-Awareness*, 2000.

[Dal82] Barrie G. Dale. The Management of Engineering Change Procedure. *Engineering Management International*, 1(3):201–208, 1982.

[Das12] Dassault Systèmes SolidWorks. mobilePDM - News: mobilePDM, provider of mobile solutions for the enterprise applications is proud to announce immediate availability of mobilePDM V 2.0 for Dassault SolidWorks Enterprise PDM. http://mobilepdm.com/en/news/detail/6, 2012. [Retrieved: 18.09.2013].

[Deu13] Deutsche Bundesbank. Basel III. http://www.bundesbank.de/Navigation/DE/Kerngeschaeftsfelder/Bankenaufsicht/Basel3/basel3.html, 2013. [Retrieved: 31.08.2013].

[Deu12] Deutsches Institut für Normung. Qualitätsmanagementsysteme - Grundlagen und Begriffe (ISO 9000:2005); Dreisprachige Fassung EN ISO 9000:2005, 2005-12.

[DG92] Alessandro Duranti and Charles Goodwin. *Rethinking context: Language as an interactive phenomenon*, volume no. 11 of *Studies in the social and cultural foundations of language*. Cambridge University Press, Cambridge [England], New York, 1992.

[DHA10] Paul Dunphy, Andreas P. Heiner, and N. Asokan. A Closer Look at Recognition-based Graphical Passwords on Mobile Devices. In *Proceedings of the Sixth Symposium on Usable Privacy and Security - SOUPS '10*, page 1. ACM Press, 2010.

[DHN02] Monica Divitini, Ove Kristian Haugalokken, and Per-Arne Norevik. Improving communication through mobile technologies: Which possibilities? In *Proceedings. IEEE International Workshop on Wireless and Mobile Technologies in Education*, pages 86–90. IEEE Comput. Soc, 2002.

[Dig09] Digi. Efficient Data Transfer over Cellular Networks - White Paper. http://www.m2mpremier.com/uploadFiles/wp_efficientdatatransfer.pdf, 2009. [Retrieved: 13.03.2013].

[Don05] Jonathan Donner. The rules of beeping: exchanging messages using missed calls on mobile phones in sub-Saharan Africa. In *55th Annual Conference of the International Communication Association: Questioning the Dialogue, New York*, 2005.

[Dor05] Richard C. Dorf. *The Engineering Handbook*. CRC Press, Boca Raton [Fla.], 2nd edition, 2005.

[DP08] David Dearman and Jeffery S. Pierce. "It's on my other computer!": Computing with Multiple Devices. In *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, page 767. ACM Press, 2008.

[Dud13a] Dudenverlag. Kontext, der. http://www.duden.de/rechtschreibung/Kontext, 2013. [Retrieved: 13.03.2013].

[Dud13b] Dudenverlag. kontextsensitiv. http://www.duden.de/rechtschreibung/kontextsensitiv, 2013. [Retrieved: 13.03.2013].

[Duf13] Thomas J. Duffy. *Programming with mobile applications: Android, iOS, and Windows Phone 7*. Course Technology/Cengage, Boston, MA, 2013.

[Edw09] Alan Edwards. PLC Engineer. http://www.plcprogrammers.com/PLCEngineer.htm, 2009. [Retrieved: 12.11.2014].

[EHTA11] Timofey Ermilov, Norman Heino, Sebastian Tramp, and Sören Auer. OntoWiki Mobile – Knowledge Management in your Pocket. In Grigoris Antoniou, Marko Grobelnik, Elena Simperl, Bijan Parsia, Dimitris Plexousakis, Pieter de Leenheer, and Jeff Pan, editors, *The Semantic Web: Research and Applications*, volume 6643 of *Lecture Notes in Computer Science*, pages 185–199. Springer Berlin / Heidelberg, 2011.

[Eig91] Martin Eigner. *Engineering Database: Strategische Komponente in CIM-Konzepten*. Praxiswissen CA.-Techniken. Hanser, München, Wien, 1991.

[Eil13] Dirk W. Eilert. *Mimikresonanz: Gefühle sehen, Menschen verstehen*. Junfermann Verlag, Paderborn, 2013.

[Eiß11] Karsten Eiß. s.Oliver setzt auf Version 6 von Dassault Systèmes: Internationales Modeunternehmen verbessert die Zusammenarbeit und erhöht die Effizienz im Produktentstehungsprozess mit Dassault Systèmes. `http://www.3ds.com/de/pressemitteilungen/einzeln/soliver-finds-the-perfect-fit-with-dassault-systemes-version-6/`, 2011. [Retrieved: 30.08.2013].

[Enk78] Gg Enke. Elektronische Datenverarbeitung bei der Produktionsplanung und -steuerung. (Band VII - Wirtschaftlichkeit von Fertigungssteuerungssystemen.) Herausg.: VDI-Gesellschaft Produktionstechnik - ADB. 133 S., 42 Abb. und 37 Tab. (1977) VDI-Verlag GmbH, Düsseldorf - kart.: 19,80 DM (f. VDI-Mitgl. 10% Nachlaß/ISBN 3-18-403078-4). *Materials and Corrosion/Werkstoffe und Korrosion*, 29(5):384, 1978.

[ES09] Martin Eigner and Ralph Stelzer. *Product Lifecycle Management: Ein Leitfaden für Product Development und Life Cycle Management*. Springer, Berlin, 2nd edition, 2009.

[ETI$^+$11] Yuta Endo, Masaharu Takahashi, Koichi Ito, Kazuyuki Saito, and Soichi Watanabe. Experimental Evaluation on SAR around the Implanted Cardiac Pacemaker by Mobile Phone. In *Proceedings of the 4th International Symposium on Applied*

Sciences in Biomedical and Communication Technologies - ISABEL Í11, pages 1–3. ACM Press, 2011.

[Eur07] European Commission. Special Eurobarometer - Electromagnetic Fields: Report. http://ec.europa.eu/public_opinion/archives/ebs/ebs_272a_en.pdf, 2007. [Retrieved: 12.05.2013].

[Eur10] European Commission. Special Eurobarometer - Electromagnetic Fields: Report. http://ec.europa.eu/public_opinion/archives/ebs/ebs_347_en.pdf, 2010. [Retrieved: 12.05.2013].

[Eva03] Eric Evans. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley, Boston, MA [etc.], 2003.

[Fin01] Karsten Finger. *Untersuchungen zur Kraftanregung durch die Verbrennung beim direkteinspritzenden Common-Rail-Dieselmotor unter Berücksichtigung des Körperschallübertragungsverhaltens*. Techn. Universität Darmstadt, Darmstadt, 2001.

[Fin13] Jon Fingas. Android tops 81 percent of smartphone market share in Q3. http://www.engadget.com/2013/10/31/strategy-analytics-q3-2013-phone-share/, 2013. [Retrieved: 07.04.2014].

[Fle05] Elgar Fleisch. *Das Internet der Dinge: Ubiquitous Computing und RFID in der Praxis*. Springer, Berlin, 2005.

[Gar13] Ariel Fredrickson Gara. Très Chic Arabia – Planes, trains, automobiles, boats and rickshaws (India adventures part 4). http://treschicarabia.com/planes-trains-automobiles-boats-and-rickshaws-india-adventures-part-4/, 2013. [Retrieved: 12.05.2013].

[Gid97] Anthony Giddens. *Die Konstitution der Gesellschaft: Grundzüge einer Theorie der Strukturierung*. Campus-Verl., Frankfurt/Main [u.a.], 3rd edition, 1997.

[GK09] Tom Gross and Michael Koch. *Computer-Supported Cooperative Work*. Interaktive Medien. Oldenbourg, München [u.a.], 2009.

[Glü10] Helmut Glück, editor. *Metzler-Lexikon Sprache*. Metzler, Stuttgart, Weimar, 4th edition, 2010.

[GM04] Victor M. González and Gloria Mark. "Constant, Constant, Multi-tasking Craziness": Managing Multiple Working Spheres. In *Proceedings of the 2004 conference on Human factors in computing systems - CHI '04*, pages 113–120. ACM Press, 2004.

[Gog06] Gerard Goggin. *Cell Phone Culture: Mobile technology in everyday life*. Routledge, London, New York, 2006.

[Gol11] David Goldman. Carrier IQ: Your phone's secret recording device. `http://money.cnn.com/2011/12/01/technology/carrier_iq/index.htm`, 2011. [Retrieved: 08.05.2013].

[Gra97] Cornelius Graack. *Telekommunikationswirtschaft in der Europäischen Union: Innovationsdynamik, Regulierungspolitik und Internationalisierungsprozesse ; mit 35 Tabellen*. Physica-Verl., Heidelberg, 1997.

[Gri06] Michael Grieves. *Product Lifecycle Management: Driving the Next Generation of Lean Thinking*. McGraw-Hill, New York, 2006.

[Gru01] Jonathan Grudin. Partitioning Digital Worlds: Focal and Peripheral Awareness in Multiple Monitor Use. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '01*, pages 458–465. ACM Press, 2001.

[Hel07] Walter Hell. *Auswirkungen der virtuellen Mobilität*. Mobilitätsverhalten in der Freizeit. Springer, Dordrecht, 2007.

[Hen12] Harvey Henao. Polytechnic Institute of New York University and AT&T: Small Business Cyber Security Survey Key Findings. `http://www.att.com/Common/about_us/pdf/cyber_security_findings.pdf`, 2012. [Retrieved: 07.03.2013].

[HFH^{+}05] Thomas Hess, Stefan Figge, Heidemarie Hanekop, Iris Hochstatter, Dieter Hogrefe, Christian Kaspar, Barbara Rauscher, Markus Richter, Andre Riedel, and Marco Zibull. Technische Möglichkeiten und Akzeptanz mobiler Anwendungen: Eine interdisziplinäre Betrachtung. *WIRTSCHAFTSINFORMATIK 47*, (47):6–16, 2005.

[HG11] Parissa Haghirian and Philippe Gagnon. *Case Studies in Japanese Management*. World Scientific, Singapore, Hackenksack, NJ, 2011.

[HHHT10] Ian Harris, Friedhelm Hillebrand, Kevin Holley, and Finn Trosby. *Short Message Service (SMS): The Creation of Personal Global Text Messaging*. Wiley, Chichester, West Sussex, U.K., Hoboken, N.J, 2010.

[HL05] Lynne Hamill and Amparo Lasen. *Mobile World: Past, Present and Future (Computer Supported Cooperative Work)*. Hamill and Lasen, [New York], 2005.

[Höl04] Johann Höller. *Internet und Intranet: Herausforderung E-Business; mit 11 Tabellen*. Springer, Berlin [u.a.], 3rd edition, 2004.

[Hol07] Harvey Holtom. *Spain*. Working and living. Cadogan Guides, London, 2nd edition, 2007.

[Hor11] Markus Horeld. Kein gutes Jahr für Blender und Ideologen - Aufstand der Jungen. `http://www.zeit.de/politik/2011-12/rueckblick-politische-trends-2011/seite-2`, 2011. [Retrieved: 14.05.2013].

[How11] Philip N. Howard. The Arab Spring's Cascading Effects. `http://www.psmag.com/politics/the-cascading-effects-of-the-arab-spring-28575/`, 2011. [Retrieved: 14.05.2013].

[HS12] Marc Halpern and Janet Suleski. Predicts 2013: Product Design and Life Cycle Management.

http://my.gartner.com/portal/server.pt?gr=dd&ref=shareSummary&resId=2258416, 2012. [Retrieved: 08.05.2013].

[HT12] Harri Holma and Antti Toskala. *LTE Advanced: 3GPP Solution for IMT-Advanced*. Wiley, Chichester, West Sussex, U.K., Hoboken, N.J, 2012.

[Hub02] Martin Huber. *Smart Dust: Trend Report 2001/2002*. Center for Digital Technology and Management, München, 2002.

[HV04] Alexander Heßeler and Gerhard Versteegen. *Anforderungsmanagement: Formale Prozesse, Praxiserfahrungen, Einführungsstrategien und Toolauswahl*. Xpert.press. Springer, Berlin [u.a.], 2004.

[IEE90] IEEE. *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries: 610*. Institute of Electrical and Electronics Engineers, New York, 1990.

[IHR13] Pourang Irani and Juan David Hincapié-Ramos. CrashAlert: Enhancing Peripheral Alertness for Eyes-Busy Mobile Interaction while Walking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*, page 3385. ACM Press, 2013.

[Int09] International Organization for Standardization. Quality management systems – Guidelines for configuration management, 2009.

[Int13] International Telecommunication Union. The World in 2013 - ict Facts and Figures. http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2013.pdf, 2013. [Retrieved: 14.05.2013].

[IOM05] Mizuko Ito, Daisuke Okabe, and Misa Matsuda. *Personal, Portable, Pedestrian: Mobile Phones in Japanese Life*. MIT Press, Cambridge, Mass, 2005.

[Jen08] Jennifer Bowcock. Apple Introduces the New iPhone 3G: Twice as Fast at Half the Price. http://www.apple.com/pr/library/2008/06/09Apple-Introduces-the-New-iPhone-3G.html, 2008. [Retrieved: 20.12.2012].

[Joh08] Kyle Johnson. *Topics in Ellipsis*. Cambridge University Press, Cambridge, UK, New York, 2008.

[Joo13] Joomla. Developing a Model-View-Controller (MVC) Component. http://docs.joomla.org/J1.5:Developing_a_MVC_Component/Introduction, 2013. [Retrieved: 17.09.2013].

[Kac11] Upendra Kachru. *India, land of a billion entrepreneurs*. Dorling Kindersley (India), New Delhi, 2011.

[Kei13] Keio Corporation. Manners and Awareness Riding the Train - Using of the Mobile Phone in the Train. http://www.keio.co.jp/english/manners/, 2013. [Retrieved: 10.05.2013].

[Kel11] Andrew Kellett. Increasing mobile device use is causing security concerns. http://ovum.com/2011/06/05/increasing-mobile-device-use-is-causing-security-concerns/, 2011. [Retrieved: 07.03.2013].

[Kel13] Leo Kelion. Google facial password patent aims to boost Android security: Users could soon be asked to pull a series of faces to unlock their Android phones or tablets. http://www.bbc.co.uk/news/technology-22790221, 2013. [Retrieved: 18.06.2013].

[KH06] Aniruddha Kulkarni and Rattikorn Hewett. Architecture for Interoperability and Reuse in Data Mining Systems. In *2006 IEEE Region 5 Conference*, pages 16–21. IEEE, 2006.

[Kin09] Lorrie Kinney. Engineering Change Management Recommendations Could Save Industry Hundreds of Millions! A global collaboration to streamline management of engineering changes for all automotive OEMs and Suppliers! http://www.aiag.org/staticcontent/press/releases/GENERAL/PR_ECM_Final.pdf, 2009. [Retrieved: 13.12.2013].

[KL99] Steinar Kristoffersen and Fredrik Ljungberg. “Making Place” to Make IT Work: Empirical Explorations of HCI for Mobile CSCW. In *Proceedings of the international ACM SIGGROUP conference on Supporting group work - GROUP ’99*, pages 276–285. ACM Press, 1999.

[KMJ+09] Amy K. Karlson, Brian R. Meyers, Andy Jacobs, Paul Johns, and Shaun K. Kane. Working Overtime: Patterns of Smartphone and PC Usage in the Day of an Information Worker. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Hideyuki Tokuda, Michael Beigl, Adrian Friday, A. J. Bernheim Brush, and Yoshito Tobe, editors, *Lecture Notes in Computer Science*, pages 398–405. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[Kol11] Tobias Kollmann. *E-Business: Grundlagen elektronischer Geschäftsprozesse in der Net Economy*. Gabler, Wiesbaden, 4th edition, 2011.

[Kos06] Kurt Kosanke. ISO Standards for Interoperability: a Comparison. In Dimitri Konstantas, Jean-Paul Bourrières, Michel Léonard, and Nacer Boudjlida, editors, *Interoperability of Enterprise Software and Applications*, pages 55–64. Springer London, 2006.

[Kra03] Uwe Kraus. *ERP-OnTo-PDM: Konzept und prototypische Realisierung einer ontologiebasierten ERP/PDM-Kopplung mittels XML-Technologie*. PhD thesis, University of Duisburg-Essen, 2003.

[Kra10] Dennis Krannich. *Mobile System Design: Herausforderungen, Anforderungen und Lösungsansätze für Design, Implementierung und Usability-Testing Mobiler Systeme*. Books on Demand, Norderstedt, 1st edition, 2010.

[Kru99] Philippe Kruchten. *Der Rational Unified Process: Eine Einführung*. Professionelle Softwareentwicklung. Addison-Wesley, München [u.a.], 1999.

[LNF11] Jakub Larysz, Martin Němec, and Radoslav Fasuga. User Interfaces and Usability Issues Form Mobile Applications. In Vaclav Snasel, Jan Platos, and Eyas El-Qawasmeh, editors, *Digital Information Processing and Communications*, volume 189 of *Communications in Computer and Information Science*, pages 29–43. Springer Berlin Heidelberg, 2011.

[Lok07] Seng Loke. *Context-Aware Pervasive Systems: Architectures for a New Breed of Applications*. Auerbach Publications, Boca Raton, Fla, 2007.

[LR10] Josef Langer and Michel Roland. *Anwendungen und Technik von Near Field Communication (NFC)*. Springer, Berlin, New York, 2010.

[Luc08] Thomas Lucka. *Mobile Games: Spieleprogrammierung für Handys mit Java ME*. Hanser, München, 2008.

[Luc12] Jeff De Luca. The Latest FDD Processes. `http://www.nebulon.com/articles/fdd/download/fddprocessesA4.pdf`, 2012. [Retrieved: 12.01.2014].

[Mad11] Lester Madden. *Professional Augmented Reality Browsers for Smartphones: Programming for Junaio, Layar, and Wikitude*. Wiley, Chichester, U.K, 2011.

[Mal83] Thomas W. Malone. How Do People Organize Their Desks? Implications for the Design of Office Information Systems. *ACM Transactions on Information Systems*, 1(1):99–112, 1983.

[MALW13] Steve Mann, Mir Adam Ali, Raymond Lo, and Han Wu. FreeGlass for developers, "haccessibility", and Digital Eye Glass + Lifeglogging Research in a (Sur/Sous)Veillance Society. In *Information Society (i-Society), 2013 International Conference*, pages 48–53, 2013.

[Man13] Katrina Manson. Africa takes lead in mobile revolution. `http://www.ft.com/cms/s/0/0846ab76-8c8d-11e2-8ee0-00144feabdc0.html#axzz2TD8rLAGd`, 2013. [Retrieved: 13.05.2013].

[Mar00] Marie-Anne Winter. Das erste Handy für zwei SIM-Karten ist da! `http://www.teltarif.de/arch/2000/kw35/s2927.html`, 2000. [Retrieved: 20.12.2012].

[Mew73] Dieter Mewes. *Der Informationsbedarf im konstruktiven Maschinenbau*, volume T. 49 of *VDI-Taschenbücher*. VDI-Verl., Düsseldorf, (1973).

[MHB92] Roger Maull, David Hughes, and Jan Bennett. The role of the bill-of-materials as a CAD/CAPM interface and the key importance of engineering change control. *Computing Control Engineering Journal*, 3(2):63–70, 1992.

[MHB10] Sara Motiee, Kirstie Hawkey, and Konstantin Beznosov. Do Windows Users Follow the Principle of Least Privilege? Investigating User Account Control Practices. In *Proceedings of the Sixth Symposium on Usable Privacy and Security - SOUPS '10*, page 1. ACM Press, 2010.

[Mic09] Microsoft. Microsoft Application Architecture Guide: Designing Mobile Applications. `http://msdn.microsoft.com/en-us/library/ee658108.aspx`, 2009. [Retrieved: 15.12.2013].

[MKG13] Alexios Mylonas, Anastasia Kastania, and Dimitris Gritzalis. Delegate the Smartphone User? Security Awareness in Smartphone Platforms. *Computers & Security*, 34:47–66, 2013.

[MPT09] Tara Matthews, Jeffrey Pierce, and John Tang. No Smart Phone Is an Island: The Impact of Places, Situations, and Other Devices on Smart Phone Use, 2009. [Retrieved: 01.06.2013].

[MS00] Steve Mann and Scott Sbihli. *The Wireless Application Protocol (WAP): A Wiley Tech Brief*. John Wiley & Sons, New York [etc.], 2000.

[MS05] José Luis Maté and Andre´s Silva. *Requirements Engineering for Sociotechnical Systems*. Information Science Publ, Hershey, PA, 2005.

[MTUK95] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented Reality: A class of displays on the reality-virtuality continuum. *Proc. SPIE*, 2351, 1995.

[Mus12] Ildar Muslukhov. How to bypass Android liveness check: Bypassing liveness check in Android Jelly Bean with a public picture from Facebook and sloppy work in Paint.NET. `https://www.youtube.com/watch?v=zYxphDK6s3I`, 2012. [Retrieved: 18.06.2013].

[MV08] Steffen Mau and Roland Verwiebe. *Die Sozialstruktur Europas*. UTB, Konstanz, 2008.

[Mxi12] Mxit Lifestyle. Mxit - Product Guide. `http://site.mxit.com/`, 2012. [Retrieved: 19.06.2013].

[Nas12] Nash Control Systems Ltd. Nash Control Systems Services: Control Engineering Consultancy. `http://www.nashcontrolsystems.co.uk/Services.html`, 2012. [Retrieved: 12.11.2014].

[Nil06] Jimmy Nilsson. *Applying Domain-Driven Design and Patterns: With Examples in C# and .NET*. Pearson Education, 2006.

[NTT13] NTT DOCOMO. About DOCOMO - Company Information. `http://www.nttdocomo.co.jp/english/corporate/`, 2013. [Retrieved: 10.05.2013].

[OB12] Tolu Ogunlesi and Stephanie Busari. Seven ways mobile phones have changed lives in Africa. http://edition.cnn.com/2012/09/13/world/africa/mobile-phones-change-africa/index.html, 2012. [Retrieved: 13.05.2013].

[Ope14] Open Automotive Alliance. New roads ahead for Android and the Open Automotive Alliance. http://www.openautoalliance.net/#press, 2014. [Retrieved: 13.10.2014].

[OS07] Antti Oulasvirta and Lauri Sumari. Mobile Kits and Laptop Trays: Managing Multiple Devices in Mobile Information Work. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '07*, page 1127. ACM Press, 2007.

[OTRK05] Antti Oulasvirta, Sakari Tamminen, Virpi Roto, and Jaana Kuorelahti. Interaction in 4-Second Bursts: The Fragmented Nature of Attentional Resources in Mobile HCI. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '05*, page 919. ACM Press, 2005.

[OWAE11] Antti Oulasvirta, Mikael Wahlström, and K. Anders Ericsson. What does it mean to be good at using a mobile device? An investigation of three levels of experience and skill. *International Journal of Human-Computer Studies*, 69(3):155–169, 2011.

[Poo11] Poorvanchal Systems Private Limited. Porchys Products & Services. http://www.porchys.com/arasiphone.html, 2011. [Retrieved: 18.09.2013].

[POS+01] Mark Perry, Kenton O'hara, Abigail Sellen, Barry Brown, and Richard Harper. Dealing with Mobility: Understanding Access Anytime, Anywhere. *ACM Transactions on Computer-Human Interaction*, 8(4):323–347, 2001.

[Pou10] Louise Poubel. Do Don't Don't Do: A world of signs. http://dodontdontdo.wordpress.com/2011/12/10/priority-seats-japan-train-station/, 2010. [Retrieved: 10.05.2013].

[Pre00] Roger S. Pressman. *Software Engineering: A Practitioner's Approach*. McGraw-Hill series in computer science. McGraw Hill, Boston, Mass, 5th edition, 2001 [i.e. 2000].

[PTC13] PTC. Verbesserungsdetails: Windchill Mobile. http://www.ptc.com/appserver/wcms/relnotes/note.jsp?&icg_dbkey=826&im_dbkey=136469, 2013. [Retrieved: 17.09.2013].

[Ray10] Rajesh Ray. *Supply Chain Management for Retailing*. Tata McGraw-Hill Education, New Delhi, 2010.

[Ree11] Matthew Reed. Press release: Africa is world's second most connected region by mobile subscriptions. http://blogs.informatandm.com/3485/, 2011. [Retrieved: 13.05.2013].

[Ret03] Ruth Rettie. A Comparison of Four New Communication Technologies. In Julie A. Jacko, Constantine Stephanidis, and Don Harris, editors, *Human-Computer Interaction: Theory and Practice (Part I)*, volume 1 of *Human Factors and Ergonomics*, pages 686–690. Lawrence Erlbaum Associates, Mahwah, New Jersey, USA; London, UK, 2003.

[RFI13] RFID Centre Ltd. Elements / Categories of Mobility. http://www.rfidc.com/docs/introductiontomobility_elements.htm, 2013. [Retrieved: 12.11.2014].

[Roz09] Peter Rozek. User Experience und mobiles Webdesign. http://www.peter-rozek.de/publikationen/user-expierence-und-mobiles-websites.php, 2009. [Retrieved: 13.03.2013].

[RR06] Suzanne Robertson and James Robertson. *Mastering the Requirements Process*. Addison-Wesley, Upper Saddle River, NJ [u.a.], 2nd edition, 2006.

[Rus11] Jon Russell. Mobile tipped to grow 60 procent in Africa passing 1 billion subscriptions by 2016. http://thenext web.com/africa/2011/11/07/mobile-tipped-to-grow-60-in-africa-passing-1-billion-subscriptions-by-2016/, 2011. [Retrieved: 13.05.2013].

[RW08] Stefan Roock and Henning Wolf. Planung mit Featurelisten. *Java Magazin*, (03/2008):126–129, 2008.

[Sch03] Jochen Schiller. *Mobilkommunikation*. Informatik. Pearson-Studium, München [u.a.], 2nd edition, 2003.

[Sch12] Caroline-Victoria Schnitzer. *Linguistische Aspekte der Kommunikation in den neueren elektronischen Medien: SMS - E-Mail - Facebook*. Grin Verlag GmbH, München, 1st edition, 2012.

[Sen09] Ulrich Sendler. *Das PLM-Kompendium: Referenzbuch des Produkt-Lebenszyklus-Managements*. Springer, Berlin, 2009.

[Sha12] Stephen Shankland. Google Glass Explorer Edition. http://reviews.cnet.com/camcorders/google-glass-explorer-edition/4505-9340_7-35339166.html?ttag=fbwp, 2012. [Retrieved: 20.12.2012].

[Sie11] Siemens PLM Software. Eine Mobilitätsstrategie für PLM-Anwender: Fundierte Entscheidungen unabhängig von Zeit und Standort. http://www.plm.automation.siemens.com/de_de/Images/Siemens-PLM-Teamcenter-Mobility-Strategy-PLM-Users-wp_tcm73-121574.pdf, 2011. [Retrieved: 11.09.2013].

[Sie13] Siemens Industry Software GmbH. Teamcenter Mobility. http://www.plm.automation.siemens.com/de_de/products/teamcenter/teamcenter-mobility/, 2013. [Retrieved: 18.09.2013].

[SM14] Kerri Shotts and Suresh Mogre. *PhoneGap 3.x Mobile Application Development Hotshot: Create useful and exciting real-world apps for iOS and Android devices with 12 fantastic projects*. Packt Publishing, 2nd edition, 2014.

[SRKH06] Kirsten Sadler, Toni Robertson, Melanie Kan, and Penny Hagen. Balancing Work, Life and Other Concerns: A Study of Mobile Technology Use by Australian Freelancers. In *Proceedings of the 4th Nordic conference on Human-computer interaction changing roles - NordiCHI '06*, pages 413–416. ACM Press, 2006.

[SSN13] Michael Andrew Sipe, Henry Will Schneiderman, and Michael Christian Nechyba. Facial recognition, 2013.

[Sta03] D. H. Stamatis. *Six Sigma and Beyond: Design for Six Sigma*. St. Lucie, Boca Raton, Fla, London, 2003.

[Sta07] John Stark. *Global Product: Strategy, Product Lifecycle Management and the Billion Customer Question*. Springer, London, 2007.

[Sta11] Statistics Bureau of Japan (SBJ). 2010 Japan Census: Population Count based on the 2010 Census Released. http://www.stat.go.jp/english/data/kokusei/pdf/20111026.pdf, 2011. [Retrieved: 12.05.2013].

[Ste02] Dan Steinbock. *Wireless Horizon: Strategy and Competition in the Worldwide Mobile Marketplace*. AMACOM and McGraw-Hill, New York, London, 2002.

[Ste10] Ralph Steyer. *Das JavaScript-Handbuch: Einführung, Praxis und Referenz*. Programmer's Choice. Addison-Wesley, München [u.a.], 2010.

[Ste12] Steffen Zimmermann. VDMA Studie Produktpiraterie 2012: Produkt- und Know-how-Schutz. http://pks.vdma.org/documents/105969/779856/VDMA+Studie+Produktpiraterie+2012/1e1408c7-05c3-4f2e-997f-4689d8f0ee3b, 2012. [Retrieved: 20.02.2012].

[Str10] Strategic Automotive product data Standards Industry Group - SASIG. Engineering Change Management Reference Process: covering ECM Recommendation V2.0.

http://www.prostep.org/fileadmin/freie_downloads/Empfehlungen-Standards/SASIG/Whitepaper_ECM_Reference_Process_V2.0.3.pdf, 2010. [Retrieved: 13.12.2013].

[SW04] Michael Schenk and Siegfried Wirth. *Fabrikplanung und Fabrikbetrieb: Methoden für die wandlungsfähige und vernetzte Fabrik*. Springer, Berlin [u.a.], 2004.

[SW11] Ulrich Sendler and Volker Wawer. *Von PDM zu PLM: Prozessoptimierung durch Integration*. Hanser, Carl, München, 3rd edition, 2011.

[SX07] Shashi Shekhar and Hui Xiong. *Encyclopedia of Geographical Information Sciences*. Springer, New York, Berlin, 1st edition, 2007.

[Taw13] Tawkon Ltd. tawkon - Cell Phone Radiation Protection. http://tawkon.com/, 2013. [Retrieved: 12.05.2013].

[Tea10] Team Transit. Transportation Culture - Week 2 - Riding the Train. http://transitculture.blogspot.de/2010/05/week-2-riding-train.html, 2010. [Retrieved: 10.05.2013].

[The09] The Economist. Mobile-phone culture: The Apparatgeist calls: How you use your mobile phone has long reflected where you live. But the spirit of the machines may be wiping away cultural differences. http://www.economist.com/node/15172850, 2009. [Retrieved: 13.03.2013].

[TP01] Steve Teixeira and Xavier Pacheco. *Borland Delphi 6: Developer's Guide*. Sams publishing, Indianapolis (Ind.), 2001.

[TRU11] TRUSTe. Smart Privacy for Smartphones: Understanding and delivering the protection consumers want. http://www.truste.com/why_TRUSTe_privacy_services/harris-mobile-survey/, 2011. [Retrieved: 07.03.2013].

[TTS14] TTS Group ASA. Services: In-Voyage repair. `http://www.ttsgroup.com/Services/In-Voyage-repair/`, 2014. [Retrieved: 12.11.2014].

[Tul07] Claus J. Tully. *Mobiler Alltag*. Springer Fachmedien, Wiesbaden, 2007.

[Twi12] Twitter Inc. Counting Characters | Twitter Developers. `https://dev.twitter.com/docs/counting-characters`, 2012. [Retrieved: 30.06.2013].

[two13] twopcharts. Countdown to 1 billion registered Twitter accounts. `http://twopcharts.com/twitter1billion.php`, 2013. [Retrieved: 14.05.2013].

[TYP13] TYPO3 Association. What is Flow? `http://flow.typo3.org/`, 2013. [Retrieved: 18.09.2013].

[Vac09] John R. Vacca. *Computer and Information Security Handbook*. The Morgan Kaufmann series in computer security. Morgan Kaufmann, Burlington, Mass, 2009.

[Val11] Bernhard D. Valnion. Durch geschicktes Anforderungsmanagement entspannter konstruieren. *economic engineering Nr. 6, Oktober/November 2011*, Oktober/November 2011(6):14–16, 2011.

[VB05] Dietmar Vahs and Ralf Burmester. *Innovationsmanagement: Von der Produktidee zur erfolgreichen Vermarktung*. Praxisnahes Wirtschaftsstudium. Schäffer-Poeschel, Stuttgart, 3^{rd} edition, 2005.

[Vel12] Chris Velazco. The Wrong Way: Path Uploads iOS Users' Address Books Without Permission. `http://tcrn.ch/w0bkql`, 2012. [Retrieved: 08.05.2013].

[VK10] Estel Vilaseca and Eva Korte. *Die Modenschau: Konzept - Gestaltung - Umsetzung*. Stiebner, München, 2010.

[Voj08] Milan Vojnovic. On Mobile User Behaviour Patterns. In *2008 IEEE International Zurich Seminar on Communications*, pages 26–29. IEEE, 2008.

[VWW11] Craig Villamor, Dan Willis, and Luke Wroblewski. Touch Gesture - Reference Cards. `http://static.lukew.com/TouchGestureCards.pdf`, 2011. [Retrieved: 27.05.2013].

[Wei91] Mark Weiser. The Computer for the 21st Century. *Scientific American*, 265(3):94–104, 1991.

[Wha09] WhatsApp Inc. WhatsApp 2.0 is submitted. `http://blog.whatsapp.com/index.php/2009/08/whatsapp-2-0-is-submitted/`, 2009. [Retrieved: 20.12.2012].

[Wor11a] World Batch Forum Staff. *ISA 88 and ISA 95 in the Life Science Industries*. Momentum Press, Highland Park, revised edition, 2011.

[Wor11b] World Health Organisation. IARC classifies radiofrequency electromagnetic fields as possibly carcinogenic to humans. `http://www.iarc.fr/en/media-centre/pr/2011/pdfs/pr208_E.pdf`, 2011. [Retrieved: 12.05.2013].

[WSV11] Anna Wasmer, Günter Staub, and Regine W. Vroom. An industry approach to shared, cross-organisational engineering change handling - The road towards standards for product data processing. *Computer-Aided Design*, 43(5):533–545, 2011.

[Yii13] Yii Software LLC. Fundamentals: Model-View-Controller (MVC) | The Definitive Guide to Yii | Yii PHP Framework. `http://www.yiiframework.com/doc/guide/1.1/en/basics.mvc`, 2013. [Retrieved: 17.09.2013].

[Yuj11] Ruby Yuji. Why Having a Baby in Japan is a Bad Idea. `http://kokusaibaby.wordpress.com/2011/01/08/why-you-should-not-have-a-baby-in-japan/`, 2011. [Retrieved: 10.05.2013].

[Zän00] Thomas W. Zängler. *Mikroanalyse des Mobilitätsverhaltens in Alltag und Freizeit*. Mobilitätsverhalten in Alltag und Freizeit. Springer, Berlin [u.a.], 2000.

[ZG10] Peter Zöller-Greer. *Software Architekturen: Grundlagen und Anwendungen*. Reihe Wissen & Praxis "kompakt". Composia-Verl., Wächtersbach, 3rd edition, 2010.

[Zsi12] Helmut Zsifkovits. Logistik. *Logistik*, 3673 : Betriebswirtschaftslehre, Technik, Ingenieurwesen, 2012.