

Abschlussarbeit an der TSU Uster im NDS Web Engineering

Thema :

BDS – Bernina Diagnostic System



Erstellt von:

Boris Zickenberg und Stephan Jehle

im

Sommersemester 2004

Betreut durch:

Stefan Zeiger

1. Abstract

The BDS (Bernina Diagnostic Software) is a tool for Bernina service technicians to alleviate her job. A technician is a person who is in charge for repairing or maintaining Bernina sewing and embroidery machines.

This includes all removed products of the 111 years company history together with the current sales palette.

Up to now, this was a mixture of mechanical experience like a Swiss watchmaker and a built in service software which has to be developed for each machine type separately.

With the enhancement of the sewing and embroidery machines to software driven sewing and embroidery systems, a new way into the future has to be developed.

For a company like Bernina, it is too expensive and needs too much manpower to develop embedded service software for each platform separately. There is a great variety in machine control hardware types from the low end mechanical machines (1008) up to the 200MHz clocked StrongArm CPU in the Artista 200.

Generalization of the service software failed not only on different CPU types. With new software development techniques like 'model driven architecture', this problem can be solved if a minimum hardware environment is present on the machine.

The other problem is the variety of screens. Here is the range between machines without screen (Overlocker) up to a high resolution colored STN screen (A200).

Because of the dependency of service functions on the screen and the CPU respectively the operating system, it is necessary to disconnect the service software from the machine.

The solution is a PC driven maintenance application together with a general database which will grow with the machine development. A troubleshooting database is the solution to achieve the knowledge of the older machines with the possibility of an upgrade for further developments.

So, we have detected two important things, a database for troubleshooting and service software for maintaining sewing/embroidery machines.

But, these things don't cover the whole problem. Every machine has a spare part catalogue which contains mechanical, electronically and software parts. For technicians it is relevant to know which spare parts will fit which machine type and are there any compatibility problems between software versions and other parts of the machine.

To cover this problem, the BDS concept was enhanced with the possibility to check the compatibility with an online query of a Bernina based compatibility database (compDB).

Summarized, the tool which can merge these three points together is a Browser. A Browser is able to establish an internet connection to open the way of connecting to a database on a server, it can use the ports of a PC to open a connection and establishing a command protocol to a machine and it supports the troubleshooting process of a question and answer sessions by hyperlinks.

This document shows the realization of these three points in a Browser environment including a description of used tools and an outlook what things Bernina has to do to enhance the possibilities of BDS in the future.

2. Expression of Thanks



Many thanks to Mr. Veltschi, owner of Bernina, who supported our studies by paying the school fees and for his morally assistance.

Also, we have to thank Mr. Haubner who gives us the chance to work on this very interesting theme for Bernina.

Jointly responsible for finishing this job is our supporter Mr. Zeiger and the clients of this project Mr. Kenyon and Mr. Hoksbergen.

Special Thanks to our layout professional Mrs. Czibulya and our advocatus diabolo Dr. Despot.

Theirs can be sure, we did our best.

Table of Content

1.	ABSTRACT	2
2.	EXPRESSION OF THANKS.....	3
3.	DOCUMENT DESCRIPTION	7
4.	DESCRIPTION OF BDS (BERNINA DIAGNOSTIC SYSTEM)	8
4.1	History	8
4.2	Presence.....	9
4.3	Conclusions	9
4.4	Features and conceptual formulations.....	12
4.4.1	Feature OHT (Offline Help for Troubleshooting) and EMQ (Error Message Qualification)	12
4.4.2	Feature MEM (Maintenance Embroidery Module)	12
4.4.3	Feature CCD (Check Configuration Data)	12
4.4.4	Feature PFS (Preparation of Future Services).....	13
4.4.5	Organizational constraints.....	13
5.	BERNINA DEVELOPMENT PROCESS (PEP)	14
6.	PHASE M00 - PRODUKTPLANUNG (PRODUCT DEFINITION) ..	15
7.	PHASE M10 – LASTENHEFT (SPECIFICATION)	16
7.1	Scenarios.....	17
7.1.1	Scenario OHT – Offline Help for troubleshooting	17
7.1.2	Scenario EMQ – Error message qualification.....	18
7.1.3	Scenario CCD – Check compatibility database.....	19
7.1.4	Scenario MEM – Maintenance embroidery module.....	20
8.	PHASE M20 – LÖSUNGSKONZEPT (TECHNICAL SOLUTION) ..	21
8.1	Process Overview.....	21
8.2	Process I – Bernina Server.....	22
8.2.1	CMS – Mambo.....	22
8.2.2	Cold Fusion (CF) Server	23
8.2.3	Spare Parts Database	25
8.3	Process II - Browser	26
8.4	Process III – Embroidery module	27

9.	PHASE M30 – PFLICHTENHEFT (SPECIFICATION)	28
9.1	Description	28
9.2	Use Case	29
9.2.1	UCBDS1: Maintain Embroidery Module (MEM) with a connected PC.....	29
9.2.2	UCBDS2: Offline Help for trouble shooting (OHT)	30
	Actors.....	30
9.2.3	UCBDS3: Check compatibility by using the compatibility database (CCD)..	31
9.2.4	UCBDS4: Error Message Qualification (EMQ)	32
9.3	Tools	33
9.3.1	Code development tools	33
9.3.2	Server services.....	33
9.3.3	Deployment tools.....	34
9.3.4	Client tools	34
9.4	Time Table	35
9.4.1	Project plan	35
9.4.2	Comparison of calculated and actually time (SIV)	37
9.4.3	Workload and consumed time	38
9.5	Workload (Arbeitspakete).....	39
10	PHASE M40 – FUNKTIONSMUSTER (FEASIBILITY)	40
10.1	Description	40
10.2	Documents	40
10.3	Risk assessment.....	40
10.4	Risk elimination	41
10.4.1	Use Case “check compatibility database”.....	41
10.4.2	Use Case “maintenance embroidery module”	42
10.4.3	Use Case “Offline troubleshooting”	43
10.4.4	Use Case “Error message qualification”	43
10.5	Testing results	44
10.5.1	Test result for Use Case “Error message qualification”	44
10.5.2	Test result for Use Case “Maintain Embroidery Module”.....	44
10.5.3	Test result for Use Case “Offline Troubleshooting”	44
10.5.4	Test result for Use Case “Check configuration database”	44
11	PHASE M50 – ALPHA PROTOTYP (PROTOTYPE)	45
11.1	Description	45
11.2	Documents	45
11.3	Problems and Conclusions.....	45
11.3.1	Java Runtime Environment on Windows 98.....	45

12	PHASE M60 – BETA PROTOTYP (FINAL PROTOTYPE).....	48
12.1	Description	48
12.2	Documents	Fehler! Textmarke nicht definiert.
13	FURTHER PROCEDURE.....	49
13.1	Description	49
13.2	Open problems.....	49
13.2.1	Java deployment.....	49
13.2.2	Content for troubleshooting	49
13.2.3	Serial port selection for embroidery firmware download.....	49
13.2.4	Mambo server setup on Bernina server.....	49
13.2.5	CMS Language support.....	49
13.2.6	Test adapter for embroidery module.....	50
13.3	Later Support	50
13.3.1	Java.....	50
13.3.2	Content management system	50
13.3.3	Cold Fusion	52
13.3.4	Link to CSM (CustomerSatisfactionManagement)	52
14	REFERENCE LIST	53
15	CD CONTENT	54

3. Document Description

This document is the complete project report of the Bernina Diagnostic System (BDS) referring to the Bernina development process (PEP).

The related documents for the BDS - project are included on the attached CD. It starts with the brainstorming in the project beginning phase and the first attempts to determine the project borders and finishes with the testing report of the running prototype.

In addition to the PEP documentation, there are some things for the TSU because this document is simultaneously a diploma theme in the subject “web engineering”.

The enclosed CD contains all documents, the source code and all other documents which are not pasted into this document.

Wherever a separate document exists, there is a reference on this document marked in the text.

The three points from the abstract:

- maintenance of PC connected sewing/embroidery machines
- offline help for troubleshooting
- compatibility check of parts by query on a server hosted database

will accompany this document. This is the “red thread” from the beginning to the end.

4. Description of BDS (Bernina Diagnostic System)

For a better understanding on the history of origins for BDS, we have to start a trip through history.

4.1 History

Bernina Nähmaschinenfabrik is a subsidiary of the FGAG (Fritz Gegauf AG) holding. Founded in 1893 from Karl Friedrich Gegauf, the company has 111 years experience in developing and manufacturing of high quality sewing machines.

In the end of the 19th century, the machines were 100% mechanically driven.



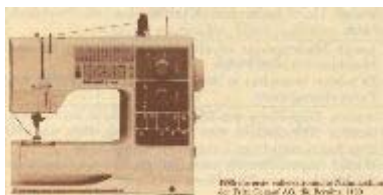
Sewing machine mechanical, 1893

To maintain and to repair this machine, the technician needs skills in mechanical adjustments like a watchmaker.

Now, a century later, this picture has rapidly changed.

The only constant thing in this period is the family of the founder. Now, Mr. Ueltschi, great grandson of Karl Friedrich Gegauf runs the company in the fourth generation.

In 1986 the first electronic sewing machine (Model 1130) was launched.



sewing machine electronically, 1986

The last part of the evolution happened 2003. The electronic, computerized sewing and embroidery system Artista 200 was launched.



sewing/embroidery system (computerized), 2003

4.2 Presence

Now, the Bernina technician has to have skills in mechanical, electronically and computer systems.

He has contact with a great variety of Bernina systems and he has to solve all customer problems.

This is more and more a mission impossible for one person in a small Bernina store who has to sell and to maintain Bernina products together.

But, maintaining the products isn't enough! He has to handle the spare parts for all products which contain a great complexity of compatibility limitations.

Another challenge is the handling of error messages which will be shown on the machine display. The message depends on the type of the screen and varies between blinking symbols on a symbolic screen of an Aactiva up to a HTML message with integrated pictures in an Artista 200.

The variety of products is summarized in the document (BDS-Current-Product-Variety.doc)

4.3 Conclusions

The most promising way to go is to build up a database with information and hints for all current and older machines.

This is the birthday of the BDS concept. BDS should support a snapshot of a knowledge base which can be deployed to all technicians all over the world.

The update of this database can be done over the internet or by delivering a CD if a technician has no access to the internet.

The possibilities of online queries on information databases in the factory over the internet and a direct connection to a sewing/embroidery system were added.

Although a clarification of error messages which are programmed in a machine with hints and suggestions for the technician were required.

The old fashion way of a built in service software is not longer an alternative. The built in software has to be run on several hardware platforms with different CPU's and operating systems.

This concept will cause a permanent work load for the software development department. The user interface has to relaunch for a new screen resolution, the control part has to be transferred on a new platform and all together has to be tested.

Another disadvantage is the impossibility of a field update. There are hundreds of thousands machine in the market. An update of these machines to implement new knowledge is very expensive and needs a long time to penetrate the market.

So, we need a flexible system which allows a fast content update over the internet to guarantee an actuality in some degrees.

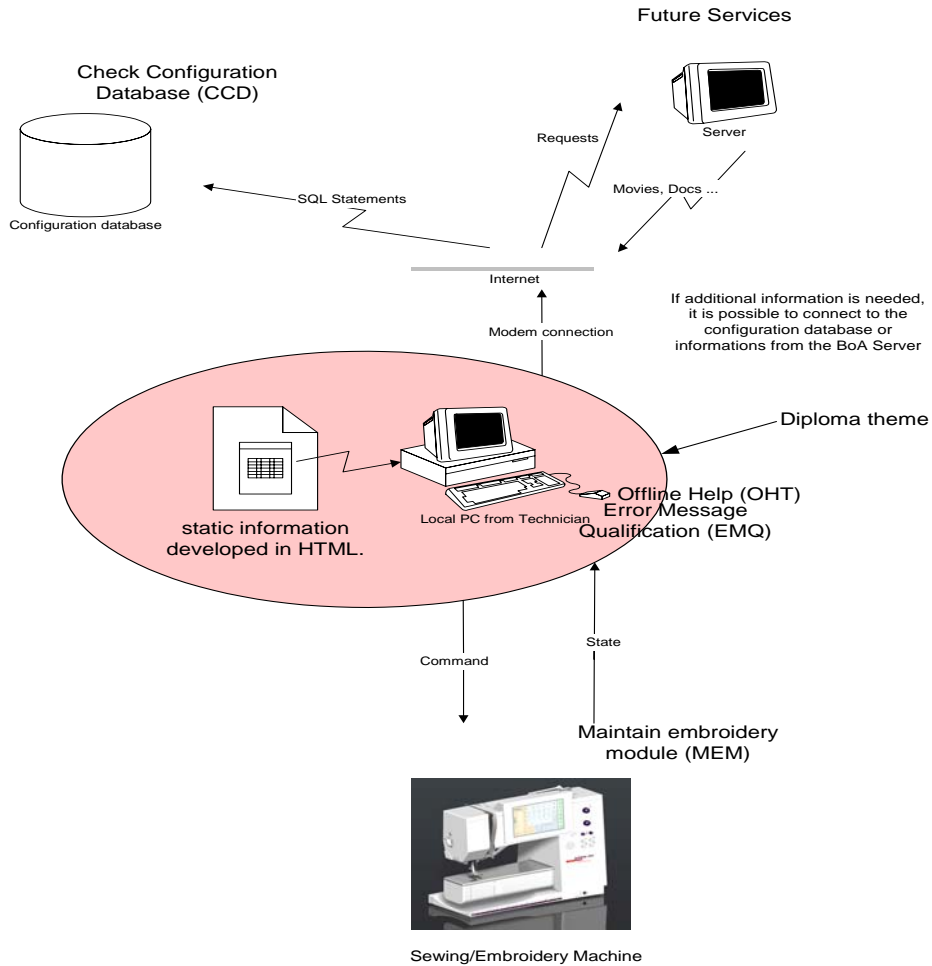
The knowledge of BDS is separated in different databases.

An existing compatibility database with the knowledge about all spare parts and software releases.

In addition there is a content management system for the online help with the summarized knowledge of all service professionals of the Bernina holding with the static extract which is located on the technician PC.

These static and dynamic parts will be shown in the picture below:

BDS (Bernina Diagnostic System)



Revision 1.0 BZ 19.01.2004

Ablage:H:\Neue_Projekte\Optima_M\Service\
Bernina-Diagnostic-System\bds-overview.vsd

4.4 Features and conceptual formulations

4.4.1 Feature OHT (Offline Help for Troubleshooting) and EMQ (Error Message Qualification)

Offline help for troubleshooting (red circle)

This is a static html part which will be deployed on a client PC to support the technician in trouble shooting. Unfortunately, not all stores dispose of an internet connection this requires an installation of a client PC. A dynamic solution was rejected because the installation of a server on every PC with every operating system is too risky. If there are problems with the installation, the average technician is overburden with a server setup like an Apache installation.

Conceptual formulation is to bundle the input documents from the technician department (After Sales Service - ASS) to a structured question and answer role-play to guide the technician to the correct solution of his problem. Responsible for the substance of the content are the ASS specialists.

The technique for OHT and EMQ is the same. The difference is the error message in EMQ. This message is the determined entry point of the offline help.

Input documents are located in folder: (Service_Input)

We've chosen a future oriented way and use a CMS system to create this content. After establishing the content, a person from the service department can maintain the content without knowledge of html or scripting languages.

4.4.2 Feature MEM (Maintenance Embroidery Module)

Direct machine connection for servicing (command <-> state exchange)

A connection to the embroidery system Aurora 430/440 is necessary because these machines are the first embroidery systems with no embedded service software included. BDS is here responsible for applying tests on these embroidery systems. This document shows the realization of the interfaces to open up the way to a universal machine interface.

Conceptual formulation is to develop an interface which enables the possibility to start functions in a browser window and to show results from the machine in the same window.

4.4.3 Feature CCD (Check Configuration Data)

Online configuration check of spare parts (SQL statements)

On the Bernina server is a snapshot of the compatibility database located which allows the technician to check the configuration of a machine. A direct connection to the SAP core database is not allowed because of security risks.

Conceptual formulation is to develop an interface which enables the possibility to start a compatibility query in a browser window and to show the results from the database in the same window.

4.4.4 Feature PFS (Preparation of Future Services)

Preparation of future services (Requests <-> Docs...)

Other service on the server can be connected in a later stage of BDS.

Because the content for troubleshooting is developed by a CMS, the same help can be offered dynamically on the server.

Another possibility is to offer some ways of eEducation. This can be done by flash animations how to adjust a main crank or by publishing movies about special themes. Because none of these things are Bernina core competences, the further development of BDS will be done externally. Later in this document, we evaluated some companies which are able to realize later enhancements.

Conceptual formulation is to prepare BDS to open up the possibility of outsourcing.

4.4.5 Organizational constraints

The development of the BDS has to follow the new Bernina “ProduktEntstehungsProzess” called PEP.

This process will be explained in the next chapter because it is important to know the signification of the milestones.

Conceptual formulation is to create all required documents for each phase.

These documents are located in folder (\Bernina-Diagnostic-System\PEP)

5. Bernina development process (PEP)

The PEP process is the adaptation of the well known software development process RUP (rational unified process) customized on the necessities of an engineering work factory.

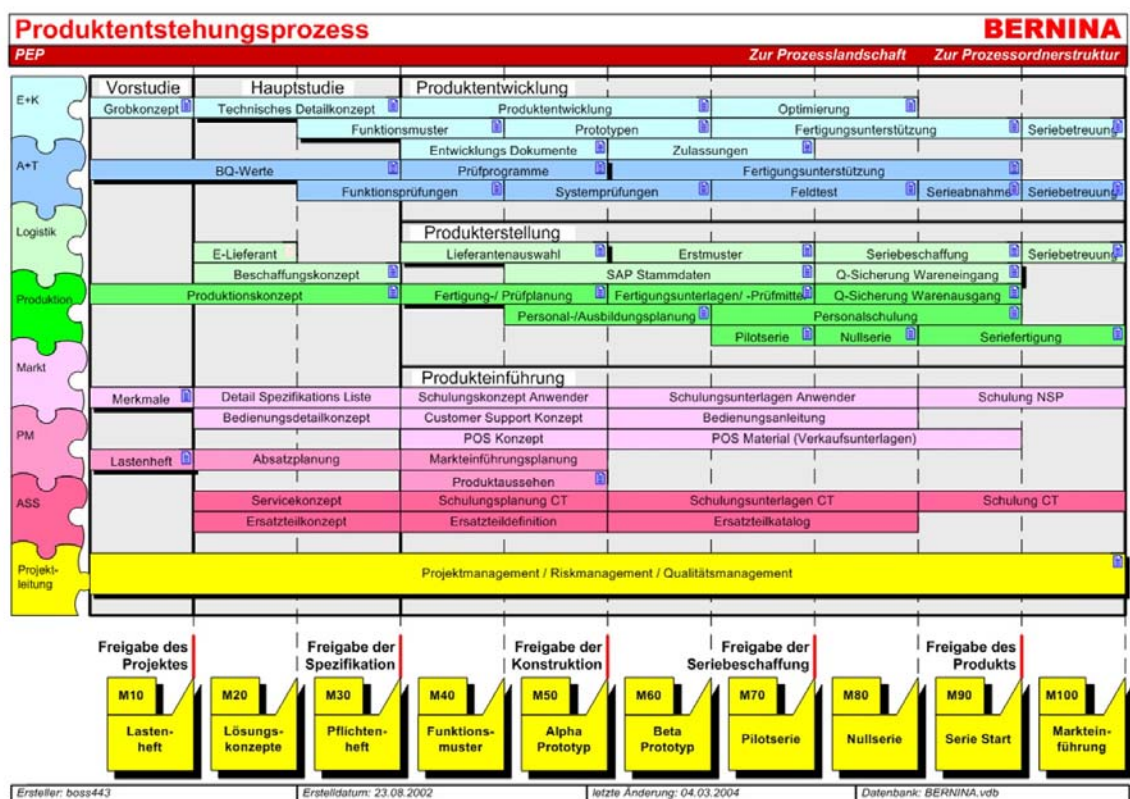
The stages of the RUP (inception, elaboration, construction, deployment) were mapped on the PEP milestones (M00 – M100):

M00-M30 is inception phase, end of this is the customer requirement specification

M40 is elaboration phase, end of this all risks are solved and an “architecture cut” is done

M50-M60 is construction phase, end of this all features are implemented

M100 is deployment, end of this is market penetration



All milestones have to be finished with a document and/or a test report with a release for the next stage.

The PEP documents and this abstract are written in English because the main stakeholder is Bernina of America.

6. Phase M00 - Produktplanung (Product definition)

In this phase the market clarifications and the description of the stakeholders and her needs take place.

This document contains also definitions and acronyms.

The related document is the vision document (M00-BDS-Vision.doc)

The most important thing is the decision to develop a PC driven service software and not to follow the way of developing embedded software for each system separately.

An additional requirement is the deployment on CD for technicians who don't have an internet connection. This requires a static extract of the troubleshooting help although there is a dynamical part generated by the CMS on the Bernina server.

7. Phase M10 – Lastenheft (Specification)

In this phase the features of the product will be finalized.

All features of the BDS are described and the development team has a clear overview about the project targets and requirements.

The related document is the Lastenheft (M10-BDS-Lastenheft.doc)

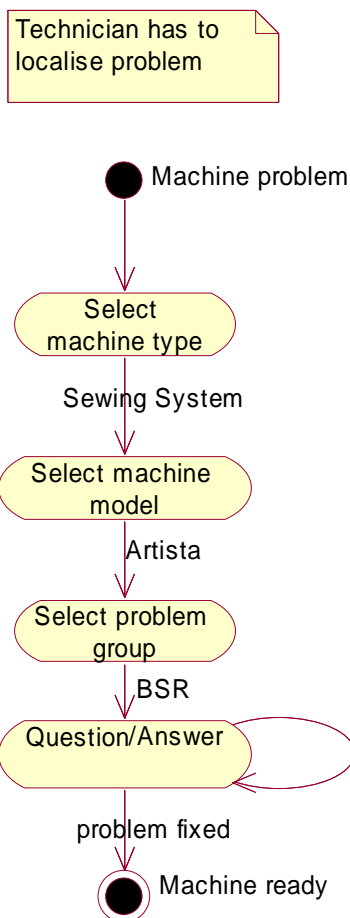
The requirements are mapped on scenarios. These scenarios containing the normal way to handle the situations for the BDS system.

7.1 Scenarios

The four possible scenarios of BDS are more detailed.

7.1.1 Scenario OHT – Offline Help for troubleshooting

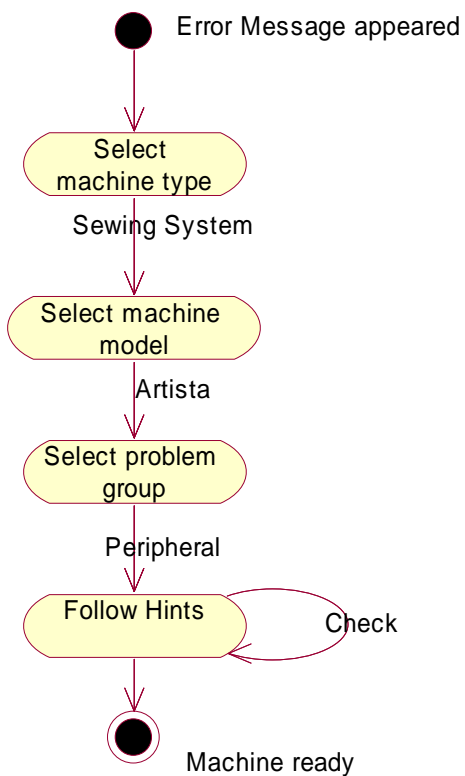
Technician detects a fault on the machine. Machine shows a behavior which does not fit the manual.



7.1.2 Scenario EMQ – Error message qualification

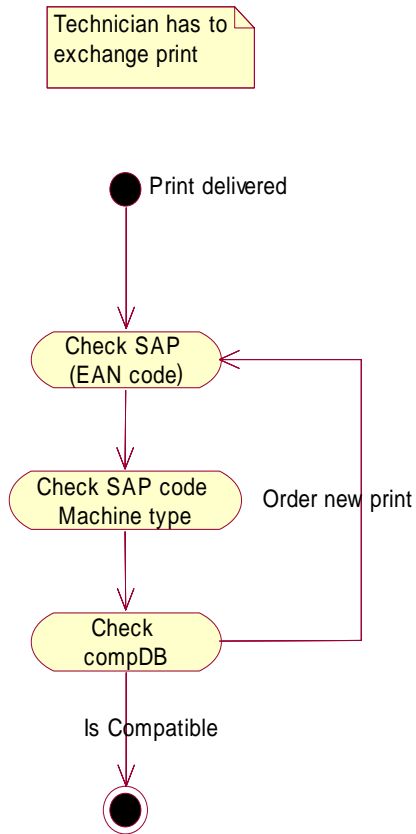
Technician has to range in an error message on the machines screen. In BDS up to now are all messages from the Artista 200 system included.

Technician has to understand error message on screen



7.1.3 Scenario CCD – Check compatibility database

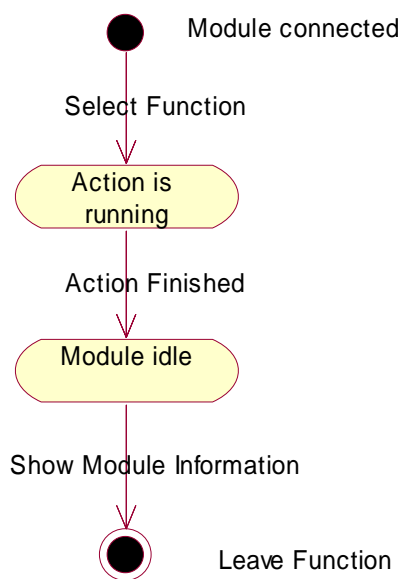
Technician has to replace a part (e.g. main print in a machine and needs compatibility check to select the correct software version.



7.1.4 Scenario MEM – Maintenance embroidery module

Technician has to maintain an embroidery module. The possible actions with all responses are listed in the PEP document (M10-BDS-Lastenheft.doc).

Technician has to maintain module



8. Phase M20 – Lösungskonzept (Technical solution)

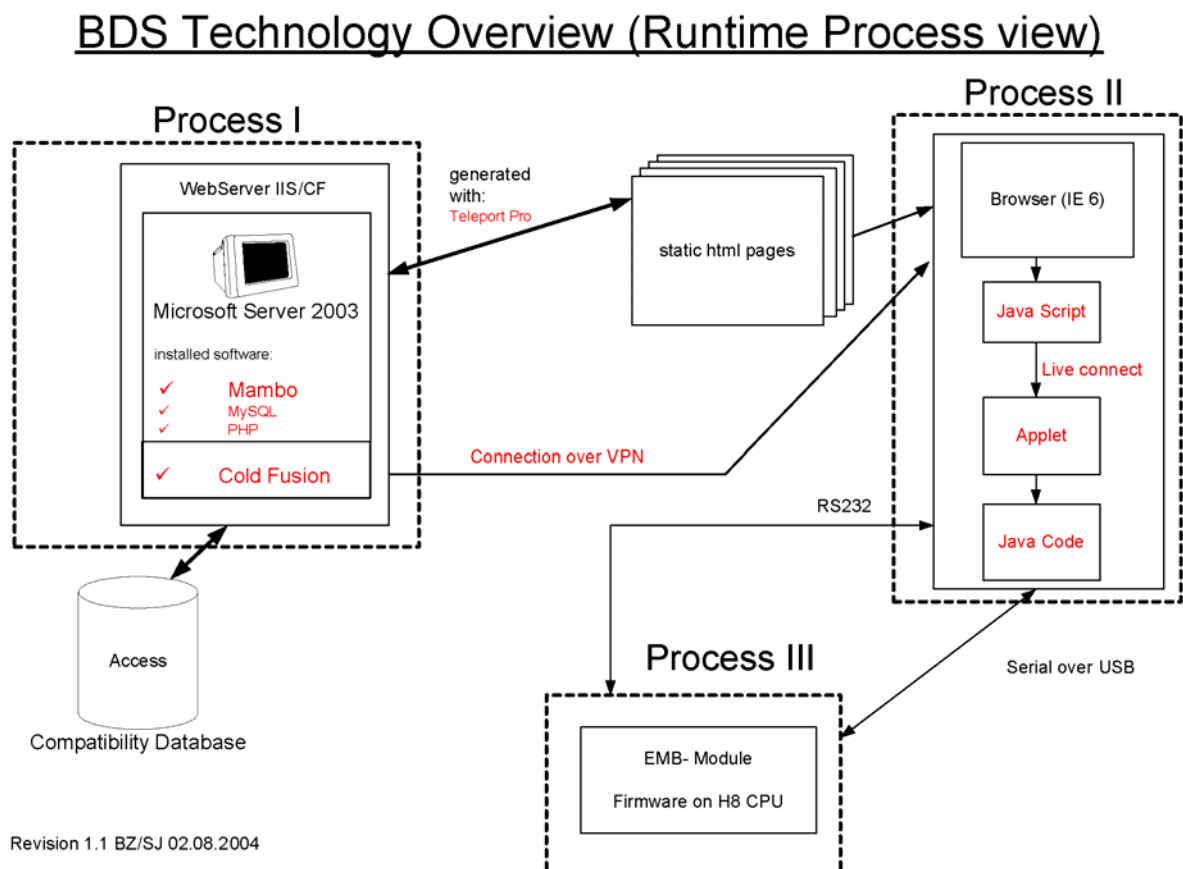
In this phase the technical solutions to implement all features of the requirement specification will be determined.

The related document is the “Lösungskonzepte” document (M20-BDS-Technische_Lösungskonzepte.doc)

The global requirements (red thread) are mapped on processes.

8.1 Process Overview

- Process I is located on the Bernina server.
- Process II is located on the client (technician PC)
- Process III is located in the embroidery module



8.2 Process I – Bernina Server

This server is a Windows 2003 IIS web server system. The configuration is strongly recommended from Bernina.

Setup and maintenance of this server is a duty from the “I” department in Bernina.

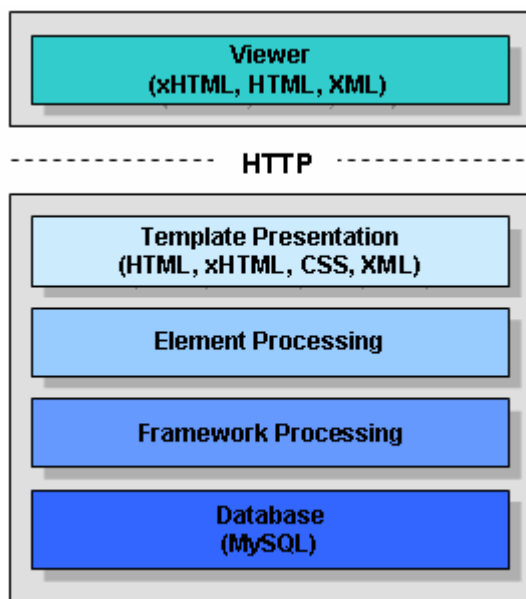
To this base setup we had to add some additional services.

8.2.1 CMS – Mambo

The CMS for the offline help and the error message qualification is managed by the Mambo CMS.

Mambo is the answer to structure the domain know how of the service department in a database.

The chosen CMS – Mambo has the following structure:

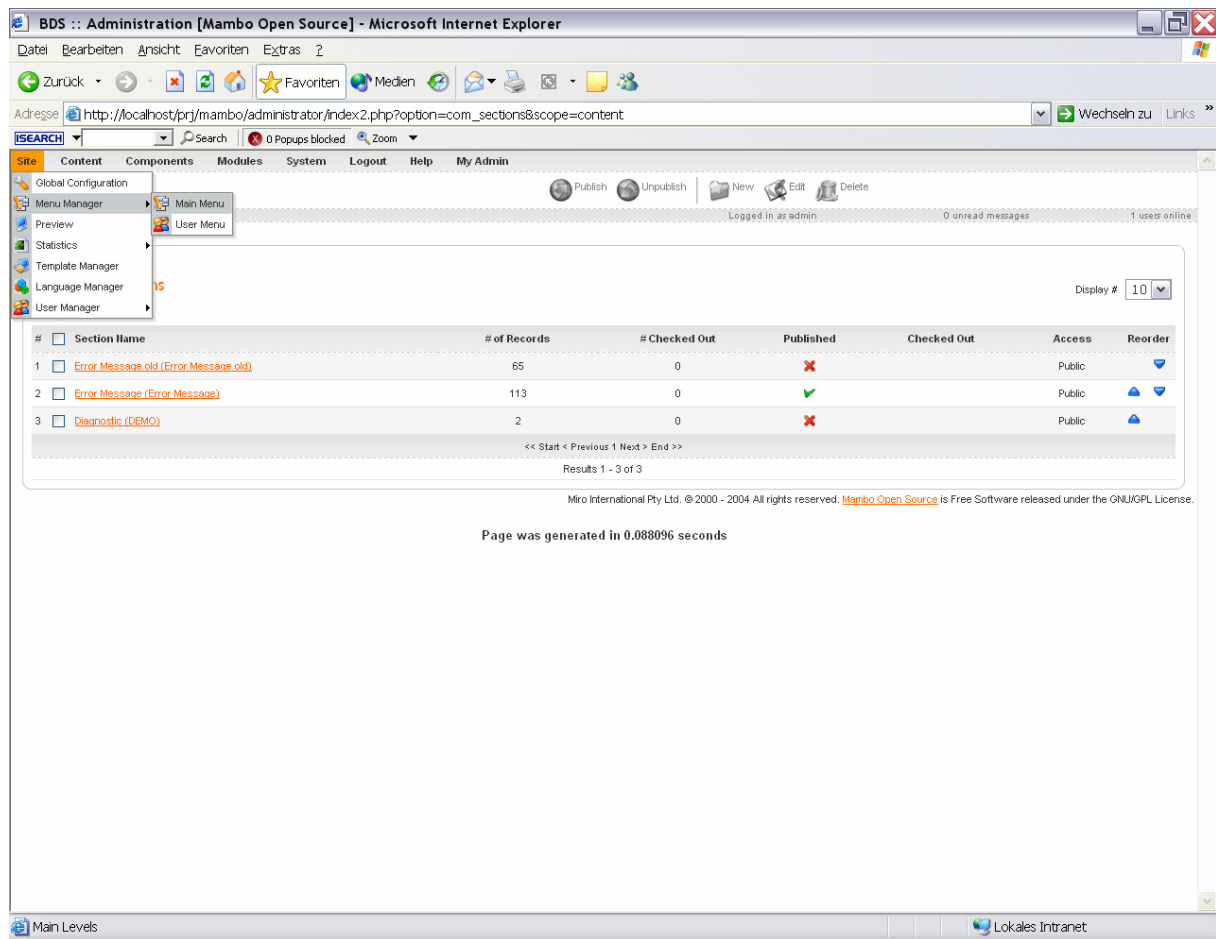


Mambo appropriates a lot of frameworks and pre defined elements like FAQ. These elements are stored in a mySql database.

After building the elements you have to choose the templates for the presentation. There are many templates defined in the mambo itself but it also possible to use a customer template. After finishing the templates and elements, the content has to be published. Now, the php pages will be loaded and the design will be run on the server.

So, the job to do is to select the suitable elements to realize the content and to fit these with our page layout.

The chosen framework for the offline help for troubleshooting was the main menu tree. The content was mounted directly in the main menu.



The error message qualification menu is built in the content area. In the menu point “categories” are the error main categories (sewing, embroidery ...).

The content of maintenance and query configuration database is not included in the mambo database. In mambo there is only a hyperlink to the entry page of this module.

All pages will be set dynamically in the CMS, for the deployment a static “extract” from the dynamic pages will be generated.

This means, all dynamic pages from the CMS will be extracted as a static HTML result set. This result set can be deployed on a CD.

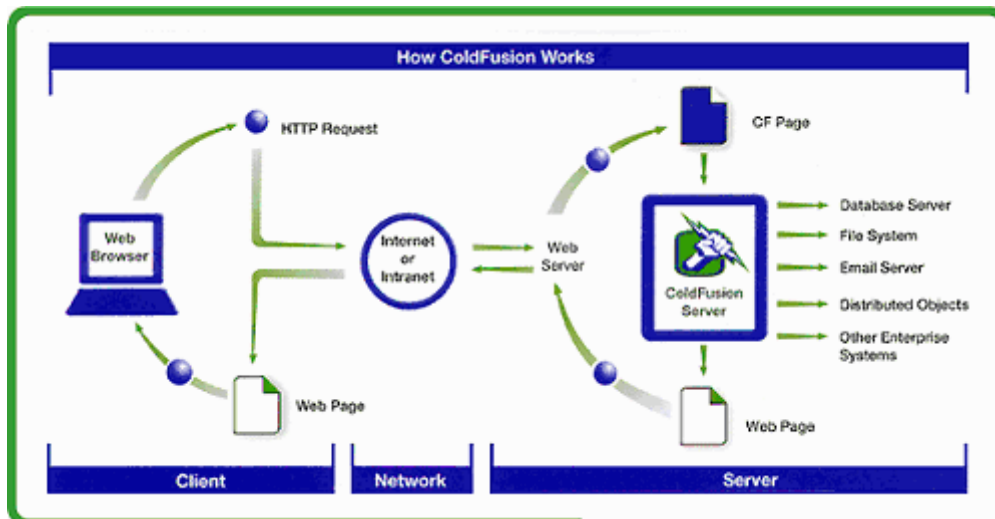
To extract the static content from the dynamic input, the software “Teleport Pro” will be used.

The teleporter will go through the dynamic tree to extract static html information. This software needs the root directory and the depth of the chosen directory tree.

8.2.2 Cold Fusion (CF) Server

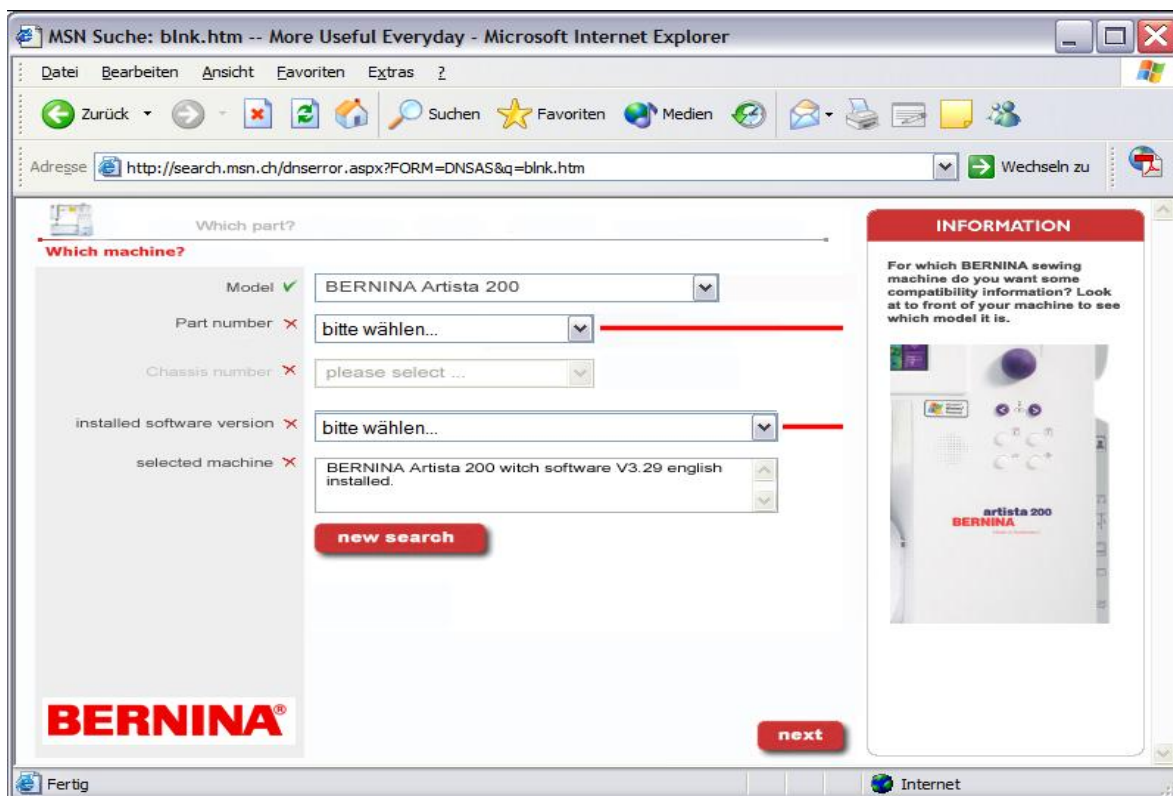
CF is a common used language which is designed to develop database connection in the Internet.

To submit dynamic queries, it is necessary to set up a web server environment where a CF server is running. This query is the answer for the check of the compatibility database (CCD).



The index script page of the server will enable the connection to the Database.

After selecting the model and the part number, the script for the compatibility check (see 8.2.3) can start.



8.2.3 Spare Parts Database

The following diagram shows the structure of the MS access compatibility database (CompDB). Creation and maintenance of this database is the BCM (Bernina Change Management) department.

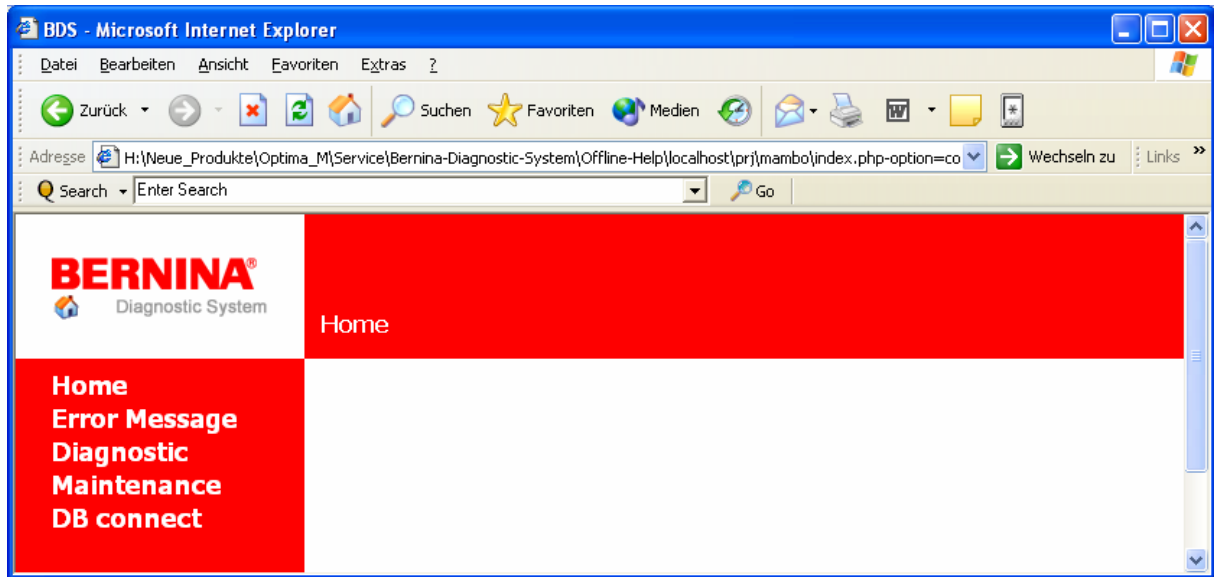
Beziehungen für COMPman
Montag, 2. August 2004



All spare parts are stored as “Components”. The Compatibility table submits the combination of two parts (foreign key Component1 and Component2). The task is to compute these inputs from the client request to detect the compatibility (element “OK” in table Compatibility).

8.3 Process II - Browser

Selected browser is the Microsoft internet explorer. Because of an alliance in marketing between Microsoft and Bernina, we had to use Microsoft products in favor. In this cases is the MS IE the first choice because the distribution in the market is nearly 90%. The browser can show html and dhtml data, so we can solve the troubleshooting concept with hyperlinks.



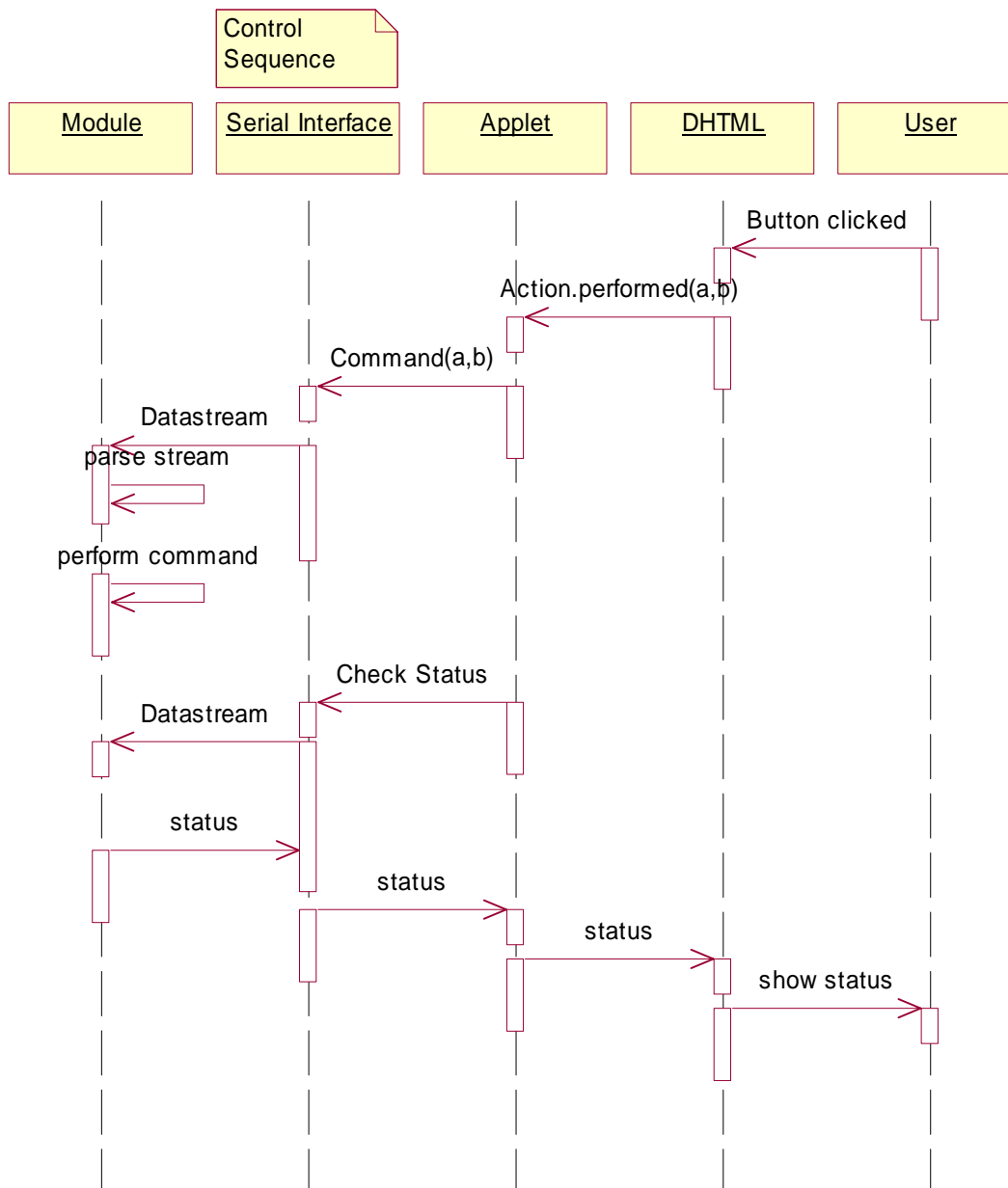
The buttons Diagnostic / Error Message / Maintenance / DB connect will link to the topics troubleshooting / error message qualification / maintain embroidery module and compatibility check.

Connection to the machine is possible with the LiveConnect technology. LiveConnect allows a hand over of parameters between JavaScript and Java applets. Additionally it is possible to call applet methods direct from java script.

To perform storing of session parameter (like actual com ports), we can use cookies.

8.4 Process III – Embroidery module

The embroidery module is a mechanical part with two stepper motors. With these motors, the module can move in x and y direction. This module has a H8 Renesas - CPU with a firmware which is responsible for the motor movements. To control the direction and the length of the steps, there is a control interface with a RS232 serial interface integrated.



9. Phase M30 – Pflichtenheft (Specification)

9.1 Description

In this phase the used technologies to implement the features will be fixed.

Final specification is done in use cases.

“Arbeitspaket”e for the project members is also included.

The time table for the project is fixed and all development and deployment tools are finally selected.

The related document is the “Pflichtenheft” document (Pflichtenheft)

This document contains the following parts:

- Use case
- Tools
- Time table
- Workload (Arbeitspakete)

To comply with the conditions of the TSU, we appended the updated time table and the comparison between scheduling and realization. A list of the consumed hours to develop the theme is also included.

9.2 Use Case

This is a description of the interactions between user and system. Every feature from the requirements is explained with a use case description.

9.2.1 UCBDS1: Maintain Embroidery Module (MEM) with a connected PC

Actors

Supporter

Brief Description

This UC describes the interaction between the SES and a connected PC.

Precondition

The PC is successfully connected with the embroidery module over a serial RS232 connection.

The module is connected to a power supply and the BDS software is running on the PC.

Main Flow

1. Supporter selects serial port on PC which is connected with them embroidery module
2. Supporter starts to download the firmware in the module with the associated button
3. Firmware will be downloaded into embroidery module
4. Firmware program reset embroidery module after download
5. Firmware starts control over embroidery module and will listen on com port
6. Supporter selects maintenance menu in BDS screen on PC
7. Supporter selects maintenance function in BDS maintenance menu on PC
8. BDS sends selected command to module
9. Module will perform selected action
10. Module will send status data to PC
11. BDS will show status information in browser windows
12. Supporter leaves BDS maintenance part.
13. Connection to machine will be disconnected by unplugging connector cable

Post condition

Machine is ready for normal operation. Serial connection was disconnected. BDS on PC is in main menu

Alternative Flow

Alternative flow A1:

Precondition:

Firmware for embroidery module was downloaded before.

Main flow steps 1 to 6 will be dropped.

Additional Information

The serial connection can be done by a serial RS232 connector cable or by an USB connection to the sewing machine.

9.2.2 UCBDS2: Offline Help for trouble shooting (OHT)

Actors

Supporter

Brief Description

This UC describes the interaction between the supporter and the BDS software on a PC to solve a problem on the SES.

Precondition

The SES is started and ready for the selected mode

Main Flow

1. Supporter selects machine platform in diagnostic menu
2. Supporter classifies problem main group (sewing / embroidery / peripheral)
3. Supporter enters main group in diagnostic menu
4. Supporter follows maintenance path by checking machine according with the hints of the BDS
5. Supporter fixes problem
6. Supporter leaves BDS diagnostic part.

Post condition

Machine is ready for normal operation. Serial connection was disconnected. BDS on PC is in main menu

Alternative Flow

Alternative flow A1:

5. Problem can't be solved:

Supporter prints pdf extract from selected BDS menu and sends this with the problem description to the CSM (customer satisfaction management)

Additional Information

9.2.3 UCBDS3: Check compatibility by using the compatibility database (CCD)

Actors

Supporter

Brief Description

This UC describes the interaction between the supporter, the BDS software on a PC and the compatibility database (CompDB) on a Bernina server.

Precondition

BDS is started and an internet connection is established. The VPN client for connecting the Bernina server is installed on the PC.

Main Flow

1. Supporter choose database menu in BDS main menu
2. A VPN (virtual private network) connection to the server where the database is located will start
3. Browser connects with the database server and shows the login-box
4. Supporter will input user name and password to establish connection
5. After verifying the authentication by the Bernina VPN termination on the firewall, the start screen of the query will appear.
6. Supporter selects the necessary parts from several drop down menus
7. Server script will send the query information to the database
7. Database will return the result set to the query script
8. Query script will show the result in the BDS browser window.
9. Supporter will end session by leaving the query script
10. VPN connection will be shutdown
11. Supporter leaves BDS database part.

Post condition

BDS on PC is in main menu

Alternative Flow

Alternative flow A1:

5. Supporter login failed because of bad authentication:

Supporter has to get authentication data from the service headquarter in Steckborn and starts with valid authentication data on main flow "1"

Additional Information

Query script is programmed in "cold fusion". The cold fusion server process is running on the bernina server.

9.2.4 UCBDS4: Error Message Qualification (EMQ)

Actors

Supporter

Brief Description

This UC describes the interaction between the supporter and the BDS software on a PC to get more information about an error message on the screen of the SES.

Precondition

The SES is started and an error message is appeared on the screen.

Main Flow

1. Supporter selects machine platform in error qualification menu of BDS
2. Supporter classifies problem main group (sewing / embroidery / peripheral)
3. Supporter selects specific error message on BDS screen
4. Supporter follows maintenance path by checking machine according with the hints of the BDS
5. Supporter fixes problem
6. Supporter leaves BDS error qualification part

Post condition

Machine is ready for normal operation.

Alternative Flow

Alternative flow A1:

3. Specific error message can't be found from supporter:

Supporter prints pdf extract from selected BDS menu and sends this with the problem description to the CSM (customer satisfaction management)

Additional Information

All error messages from the machines are included in the BDS.

9.3 Tools

There are some tools to select. We made groups to differentiate coding tools from services and others.

9.3.1 Code development tools

Part	Tool	Manufacturer	Comment	web page
Java	JDK 1.4	SUN	Development kit	www.sun.com
Java	Netbeans	IBM	Workbench V. 3.5.1	www.netbeans.org
UML	Rose 2000e	Rational	V 2.10	www.rational.com
HTML	Dreamweaver	Macromedia	Code V7.0	www.macromedia.com
HTML	Photoshop	Adobe	Design V 7.0	www.adobe.com
Code Administration	Source Safe	Microsoft	V6.0	www.microsoft.com

9.3.2 Server services

Part	Tool	Manufacturer	Comment	web page
Operating System	Windows Server 2003	Microsoft	Bernina Standard	www.microsoft.com
Remote tool (internal access)	VNC	University of Cambridge	Royalty free	www.realvnc.com
VPN (external Access)	VPN Secu - remote	Checkpoint Ltd.	Bernina Standard	www.checkpoint.com
Script Server (Cold Fusion)	Cold Fusion Server	Macromedia	V 6.1.0	www.macromedia.com
CMS	Mambo	Miro Int. Pty.	Royalty free	www.mamboserver.com
	PHP	PHP	Royalty free V 4.3.6	www.php.net
	mySql	mySql	Royalty free V 3.2.3	www.mysql.com

9.3.3 Deployment tools

Part	Tool	Manufacturer	Comment	web page
Description	Robo Demo	Macromedia	V 5.0	www.macromedia.com
Installation	InstallShield	InstallShield Inc.	V 7.01 professional	www.installshield.com

9.3.4 Client tools

Part	Tool	Manufacturer	Comment	web page
USB Driver for EMB module	FTDI	FTDI	driver for windows systems	www.ftdichip.com
Java	JRE 1.4	SUN	Java Runtime Environment	www.sun.com

9.4 Time Table

The time table contains the original schedule with the tracking remarks (grade of implementation in percent) and an analysis of the divergences (SIV SollIstVergleich). The divergences are sorted by the PEP phases.

9.4.1 Project plan

Link to document: [Microsoft project document](#)

In the top part there are only milestones from the PEP process.
In the bottom part is the realization track of the “Arbeitspakete”.

If there is a job which was not completed this will be highlighted by red color.

9.4.2 Comparison of calculated and actually time (SIV)

There are some points which had a delay in the realization.
In following there is a compilation of the greatest mistakes relating to the project phases.

9.4.2.1 Phase M00 Produkteplanung (product defining)

The vision document is a prolonged thing because some departments have to work together. The most dangerous thing is the definition of features with solutions. Development engineers are dependent on the exact definitions of a problem but sometimes the engineers from other departments are thinking about possible solutions and forgot the description. This can cause a big miss in the final product.

9.4.2.2 Phase M10 Lastenheft (requirements)

Unfortunately, there is no English style sheet for the “Lastenheft”. The danger is, that at the end of the project all documents had a different style.

9.4.2.3 Phase M20 Lösungskonzepte (solution)

The technical solution concept document had a delay of three weeks. It is impossible to check the solutions without developing a running prototype. So, the M20 and M40 phases are having a lot of intersections.

9.4.2.4 Phase M30 Pflichtenheft (specification)

The specification document was much more amount of work than we realized before. It is necessary to break the project into smaller pieces (Arbeitspakete) to open the possibility of allocation of jobs to persons. The acceptance of the use cases and the granularity of the “Arbeitspakete” devoured two non scheduled weeks.

9.4.2.5 Phase M40 Funktionsmuster (feasibility)

This phase was on time because we expected some problems with the Java Interface. The most important thing was the unusable SUN port driver and the impossibility to use client side LiveConnect from Java to JavaScript in a Microsoft Browser environment.

9.4.2.6 Phase M50 Alpha Prototyp (function prototype)

Here, we missed some vital information from the customer. Because of the fixed delivery date to the TSU, we decided to freeze the content to open it up in a later stage.

9.4.3 Workload and consumed time

This is a description for the TSU how much time we need to work on the defined packages.
We spent the weekends and some afternoons/nights during the labor week.

Package AP	Name	load / hours	comment
1	BZ	65	
2	BZ	50	
3	SJ	45	
4	SJ	15	With "T" department
5	SJ	42	
6	SJ	21	
7	BZ	12	
8	BZ	6	
9	BZ/SJ	20 20	

Summary :

Jehle 143h
Zickenberg 153h

9.5 Workload (Arbeitspakete)

Jobs do be done were separated for each project member.

Link to folder: Workload

Packages :

Responsible Zickenberg

AP-BDS-0001-ProjectManagement.doc
AP-BDS-0002-EmbroideryModule.doc -> MEM
AP-BDS-0007-InterfacePreparation.doc
AP-BDS-0008-Deployment.doc

Responsible Jehle

AP-BDS-0003-OfflineHelp.doc -> OTS
AP-BDS-0004-ServerSetup.doc
AP-BDS-0005-QueryDatabase.doc -> CCD
AP-BDS-0006-ErrorMessage.doc -> EMQ

Responsible Zickenberg , Jehle together

AP-BDS-0009-Testing.doc

10.4 Risk elimination

The detected risks from A1 to D1 referring to the risk list were explained with the technical solutions.

10.4.1 Use Case “check compatibility database”

Here comes a collection of points in the server environment.

10.4.1.1 A1 – Cold fusion server not allowed

Up to now, Bernina has a Microsoft only server environment. But in the Aurora embroidery project we started a macromedia tool chain with the director and Dreamweaver. So we could argument with a new, closed tool chain for internet projects. We will use cold fusion as scripting engine, Dreamweaver for html coding and director for eEducation.

10.4.1.2 A2 – External server access not allowed

After a meeting with the informatics department we could establish a safe VPN connection with a termination in the firewall.

10.4.1.3 A3 – External access on SAP not allowed

A corruption of the SAP database with all manufactured parts since over 100 years will cause a deadly damage to the company. To be sure, there is a daily backup on tape drives which will be weekly stored in a bank safe. But a reload of older data will cause some troubles and a stop of production.

To avoid this problem we made a snapshot of the SAP database into a Microsoft access database. This snapshot is located on the Bernina server and can be accessed by out CF scripts.

10.4.2 Use Case “maintenance embroidery module”

Here comes a collection of points in the connection between client PC and embroidery module.

10.4.2.1 B1 – Java security problem

An access from an applet to a physical resource will cause a security exception. This can be avoided when the applet is signed as a trustworthy code. This was done by using a key from a “trust certificate” company.

The steps to create the key pairs for signing applets are detailed in the PEP document.

10.4.2.2 B2 – Java serial driver

The original driver to access serial ports from Sun is not suitable. We found a link in a newsgroup the bug in this package is known since years and Sun will not fix this. The proposal from the java forum is to exchange the Sun package with a package from IBM. We did it and it worked beautiful.

10.4.2.3 B3 – Firmware download

The code to download the firmware into the embroider module is a native executable. We can't change this. Unfortunately the program can only handle com port one and three. We made a workaround with a shortcut where we can change the port selection. If the deployment phase will show a problem with port three, we have to launch a change request for this executable.

10.4.2.4 B4 – Connection to embroidery module

For the technician it is more comfortable to check the embroidery module with a connected machine. For the customer is this a great disadvantage because he has to leave both parts at the shop during the maintenance period. Our prototype will run with a special serial adapter between PC and embroidery module. The switch to use the virtual com port driver with serial over USB is prepared. If the management will claim this connection, it will be done in one or two days.

10.4.3 Use Case “Offline troubleshooting”

Here comes a collection of points for the offline help for troubleshooting. It affects the content and the mechanism to publish the content on the client PC.

10.4.3.1 C1 – Content from After Sales Service (ASS) department

The quality of the content is not at a high level. The conclusions are not always understandable.

Many points are filled with default stuff which doesn't fit the problem.

We filled in the delivered content to proof the mechanism but this has to be reworked before publishing.

10.4.3.2 C2 – Mambo installation not allowed

Mambo is an open source content management system (CMS). It is not a part from the Microsoft or Macromedia tool chain. Also it requires a mySql database and a PHP scripting support on the server.

To finish the TSU theme, we installed the CMS on a notebook. The decision for a Bernina server installation is still pending. Maybe it is possible to separate the CMS server to a company which is also responsible for the CMS maintenance.

10.4.3.3 C3 – Generation of static content

With a “ripper tool” which is able to extract static pages from a dynamic script, the problem is easily solved. This tool is the “teleport pro” tool.

10.4.3.4 C4 – Language selection

The delivered content from ASS was in German language. This is not usable to deliver it to Bernina of America (BoA). The decision if Bernina will support a single language installation or a complete installation with an online language switch is still open.

10.4.4 Use Case “Error message qualification”

It affects the content which has to be published to perform this use case.

10.4.4.1 D1 – Content from After Sales Service (ASS) department

No content was delivered. Because we were members of the Artista 200 development team, we made a content description by ourselves.

10.5 Testing results

All use cases were successfully tested. Here are the results mapped on the use cases.

10.5.1 Test result for Use Case “Error message qualification”

See document Feasibility-Test-EMQ.xls

10.5.2 Test result for Use Case “Maintain Embroidery Module”

See document Feasibility-Test-MEM.xls

10.5.3 Test result for Use Case “Offline Troubleshooting”

See document Feasibility-Test-OTS.xls

10.5.4 Test result for Use Case “Check configuration database”

See document Feasibility-Test-CCD.xls

11 Phase M50 – Alpha Prototyp (Prototype)

11.1 Description

In this phase the system will be merged first time.
The single feasibility parts have to work together in a closed system.

The installation will be done manually to check the system installation specialties.
This is a condition to build up the automatic installation which will be done in the Beta phase.

Further conclusions and appeared problems are also mentioned in this part.

11.2 Documents

The related document is the platform definition on which tests taken place:

Device-Description-Alpha-Test.doc

And the testing results:

Alpha-Test.xls

There is no bit PEP document like the phases before. Starting with the alpha prototype the most important documents are release notes and test result documentation.
Because we had no releases, we had only a test documentation of the running prototype.

11.3 Problems and Conclusions

One additional problem occurred which was not detected in the feasibility phase. The deployment of the MEM module on a windows 98 environment wasn't successful.

11.3.1 Java Runtime Environment on Windows 98

Installation of the Java Engine on the Win98 notebook wasn't successful.

To check out this point we developed a Java Runtime test applet:

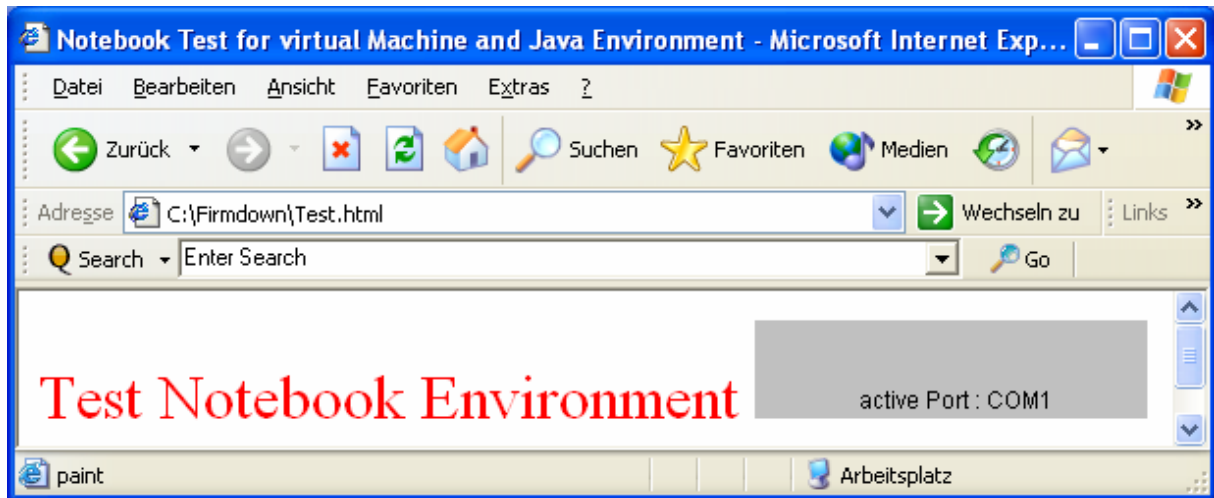
HTML code :

```
<applet code="notebookTest.class" width="200" height="50">  
    <param name="strCommand" value="COM1">  
</applet>
```

JAVA code :

```
public class notebookTest extends java.applet.Applet {  
  
    String strPort = "unknown port";  
    String[] myCommand = { "COM1", "COM3" };  
  
    public void init()  
    {  
        myCommand[0] = getParameter("strCommand");  
  
        strPort = myCommand[0];  
    }  
  
    public void paint( Graphics g )  
    {  
        showStatus("paint");  
  
        g.drawString(" active Port : " + strPort, 50, 45 );  
    }  
}
```

If the Java runtime environment works correctly, the following output will be shown:



Otherwise, the gray output field from the “draw” method is empty.

On the win98 notebook the method never starts. Possible reason is a problem with the java plug in installation for the Microsoft Internet Explorer.

Before delivering the BDS package, we have to solve this problem.

12 Phase M60 – Beta Prototyp (Final Prototype)

12.1 Description

In this phase the system will be finally merged.
All parts have to be run in an integrated environment.

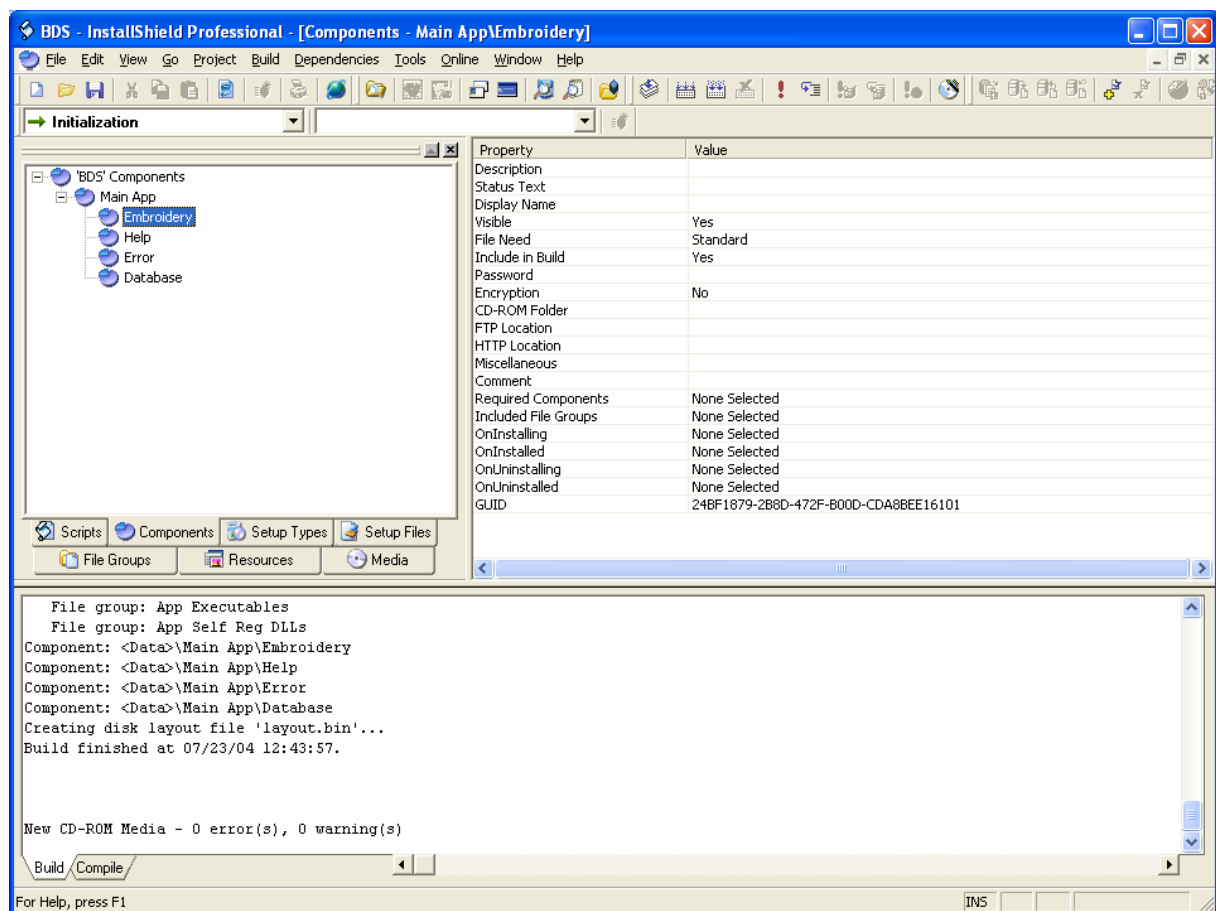
The installation will be done automatically with an Install Shield installation routine.

**This phase is not finished because of the occurred problems of the alpha phase.
Setup routine on disk will not work. Only the skeleton is prepared and tested.**

12.2 Setup

Project location of InstallShield setup routine in setup folder: [Link](#)

Oversight of Setup content:



Setup routine will install services on drive c:\.

13 Further Procedure

13.1 Description

Here comes the final conclusion. What things are good practice for the future development, which problems has to be solved and which way should Bernina go to build up a better service environment for all technicians.

Because of the necessity to outsource the maintenance of BDS, we evaluated suitable companies which can be hired for this job.

13.2 Open problems

Some things are not finally developed. In the next stage of the project, we have to fix this together with ASS department.

13.2.1 Java deployment

Java runtime environment has to be set successfully on the windows 98 notebook.

13.2.2 Content for troubleshooting

We have to integrate a complete set for troubleshooting. All parts (sewing, embroidery, peripheral) have to be covered.

13.2.3 Serial port selection for embroidery firmware download

The native code allows a firmware download only on ports com1: and com3:. Because, the virtual com port driver can map to another port, this has to be changed to allow a free port selection.

13.2.4 Mambo server setup on Bernina server

Mambo doesn't run on Bernina server. This has to be fixed. It looks like a problem with the localhost settings for security reasons.

13.2.5 CMS Language support

The way of inserting language dependent content has to be defined. Do we need a language switch in the delivered content; this implies a delivery of all languages in an installation. Or should we deploy a separate installation for each language.

13.2.6 Test adapter for embroidery module

The decision about a hardware adapter is still open. If we decide to connect only with the sewing machine as connector, the customer has to bring always the sewing machine to the dealer if he has a problem in embroidery.

13.3 Later Support

Because, the building of web based content is not a core technology for Bernina, the support has to be outsourced.

Here is the extract from our experiences with the used technologies.

13.3.1 Java

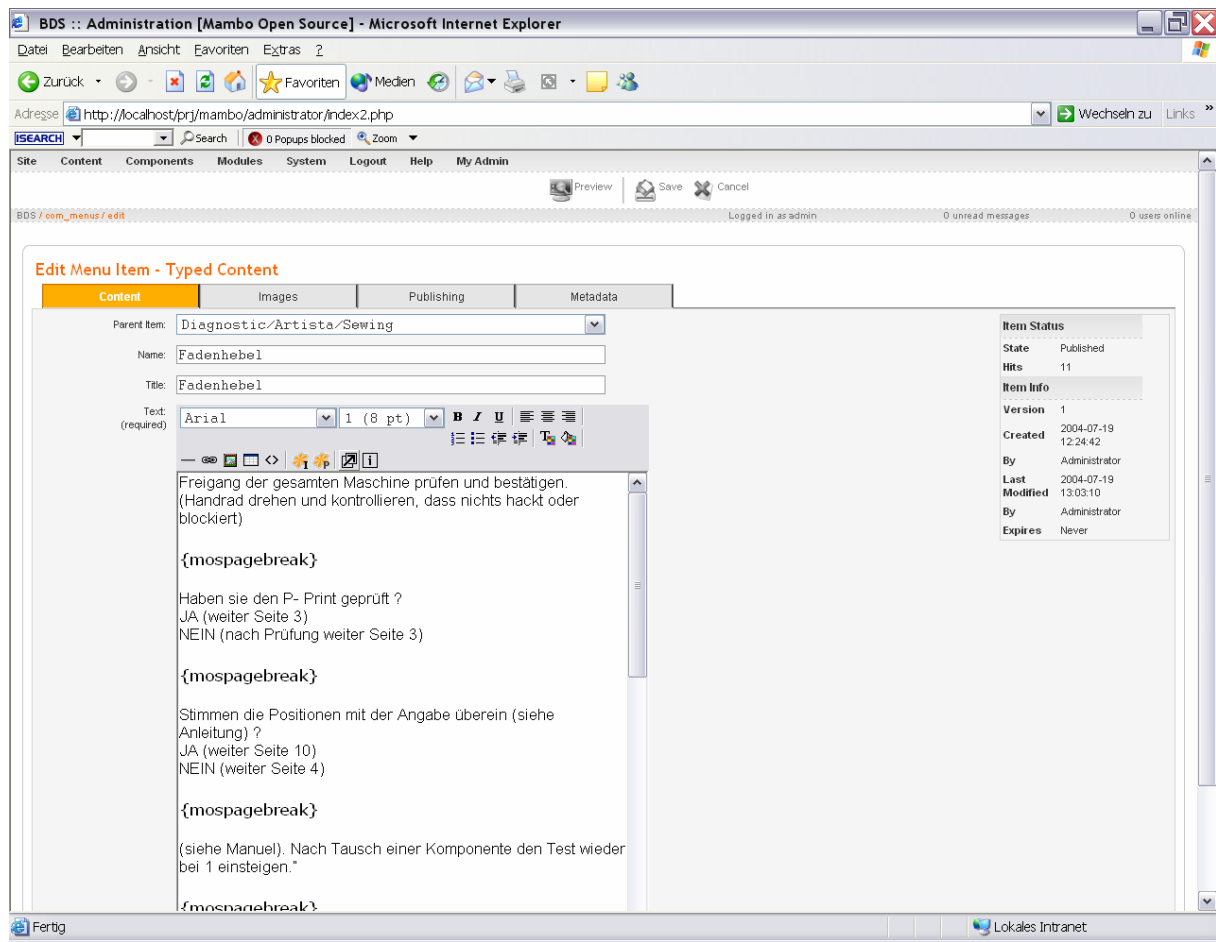
The java technology showed some problems in the deployment on different platforms. So, the alternative C# has to be analyzed. This needs a .NET runtime environment too, but the experience in deploying this environment will be checked in the Aurora project. Addicted to these results, a change in this part is necessary. Unattached from the results, the development team from Bernina is starting a C# education at the TSU.

13.3.2 Content management system

Mambo is a very powerful CMS but it needs some attendance for version updates and certain effort in adapting of the built in frameworks to fit the needs of Bernina.

The most important thing is the adaptation from the specialist's documents in an Excel sheet into the CMS.

Now, it is necessary to use the copy/paste sequence into the page:



This is a very error-prone method! We need an automatic conversion from the excel sheet into the mysql database from the mambo.

We suggest assigning Mr. Michael Janis (www.janis.ch) with the maintenance of the CMS. He is an expert in Mambo installations with a good reputation.

13.3.2.1 Skip offline development

Also, we think the way to develop an offline database is a dead end street. Bernina should perform internet equipment for all technicians.

The benefits are:

- same content for all technicians
- current content is up to date
- pdf or other conversions included in mambo server
- automatic eMail generation possible
- logging possible (to improve logical structure)
- CSM (CustomerSatisfactionManagement) database connection (see 13.3.4)

Disadvantages:

- needs 24h server up guarantee
- denial of service problems

We are sure, the benefits are more important.

13.3.3 Cold Fusion

For developing runtime queries on a database, this tool is very suitable.

To develop new scripts we investigated two companies which are qualified in taking over this job.

Alternative A: Artack Web Lab (www.artack.ch), well known from the Aurora project and familiar with the Macromedia tool chain.

Alternative B: One Byte (www.onebyte.ch), an internet development company with high skills in Cold Fusion development.

The decision for the later development will be forced by the after sales service (ASS) together with R&D department.

13.3.4 Link to CSM (CustomerSatisfactionManagement)

In the final stage, BDS should link to the currently installed Bernina CSM (Customer Satisfaction Management). CSM is a web based solution to collect problems from dealers and technicians. The solutions for the problems are stored with a CSM-ID. The CSM-ID contains the machine type and the machine part (e.g. Artista peripheral).

To link the BDS way of troubleshooting with CSM-ID, a method is needed which converts the BDS “pathway” into a valid CSM-ID. This should replace the page “contact CSM” which appears if a problem can’t be solved with the hints from the BDS troubleshooting content.

The reverse way is also useful. If CSM has a solution for a new problem which is not a part of the BDS database, it should be possible to add this solution into the Mambo – CMS database.

14 Reference List

Java development

[1] JavaScript – Rezepte, Christian Wenz, ISBN 3-934358-149-x

Server

[2] Verwalten einer Microsoft Windows Server 2003 – Umgebung, Dan Holme, ISBN 3-86063-931-5

Design

[3] Corporate Design Manual, Bernina Ausgabe 2004

[4] Webdesign, TSU Uster Dozent Janis,

Software general

[5] Software Testing, Ron Patton, ISBN 0-672-31983-7

15 CD content

Enclosed CD contains the following directories:

15.1 A200 Messages

This is the summary of all Artista 200 error messages. This is the base for the error message qualification menu.

15.2 CP_VPN_Client

This is the VPN client software for attaching an external client to the Bernina BDS server.

15.3 EMB-Module

Description documents about the interface of the embroidery module and the corresponding source code.

15.4 Offline-Help

Complete source code and static html pages for the offline help for troubleshooting.

Also included is the Java source code for maintaining embroidery module.

Entry Page for BDS is <\\Offline-Help\localhost\prj\mambo\index.php.htm>

15.5 Feasibilities

Contains the native code for firmware download in the subfolder FirmdownANS and the first suggestion for the BDS main menu.

15.6 IBM-Driver

Contains the serial port driver package for Java developed by IBM.

15.7 InstallShield

Contains the install shield project to setup the BDS application on client PC.

This is not finished and can't be used.

15.8 Mambo CMS

Contains the official mambo reference book.

15.9 Operating-System

Contains description of operating system dependent registry keys which we have to notice during installation.

15.10 PEP (Produktentstehungsprozess)

Contains all necessary documents for this process with all “Arbeitspakete” in the subfolder. Folder /Test contain the testing result of the dedicated phases.

15.11 RoboDemo

Contains the license key to use the tool. Development of the installation help is not finished.

15.12 Rose

Contains the Rational Rose files for the BDS use cases in UML.

15.13 Server

Contains the VNC remote tool, a windows2003 server reference book and sample content for eEducation.

15.14 TSU

Contains this document and official documents from the TSU.

15.15 Verisign

Contains the code to enable the possibility for creating signed applets by Bernina.

15.16 Scripts

Contains the cold fusion scripts for database queries

15.17 Service Input

Contains the input from the ASS specialists

15.17 Java

Setup for Java Runtime Environment