

Kurzübersicht: LineMod-Demoapplikation

1	Übersicht	1
2	Datenbasis BDE/Simulation	2
2.1	PDA Data Base	2
2.2	Simulation Data	2
3	Diagnostic Data Base	3
3.1	configuration & parameter	3
3.2	diagnostic data	4
3.3	batch information.....	5
4	Gantt tool	5
5	Diagnosealgorithmen	6
5.1	model based diagnosis	6
5.2	recursiv diagnosis	6
6	diagnosis result.....	6
7	demo visualisation.....	7

1 Übersicht

Die LineMod-Demonstratorapplikation ist ein Ergebnis des Forschungsprojekts LineMod. Darin wurden die Grundlagen geschaffen, mit Hilfe von Betriebsdaten liefergradsmindernde Faktoren in Getränkeabfüllanlagen automatisch zu analysieren. Die Demoapplikation ist eine beispielhafte Umsetzung der Ergebnisse, um die Machbarkeit in der Praxis zu demonstrieren.

Die Struktur der Demoapplikation des LineMod-Projekts ist in Abbildung 1 dargestellt.

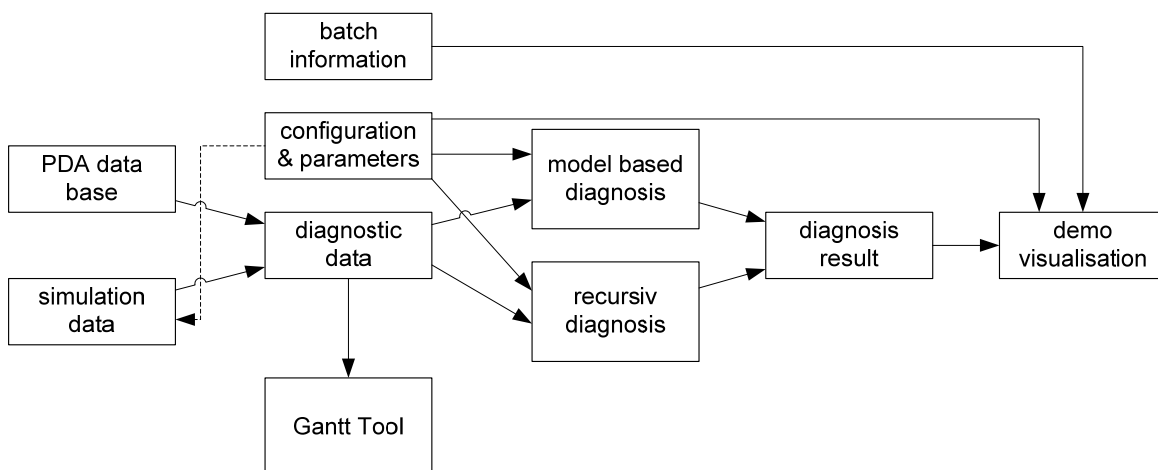


Abbildung 1: Struktur Demoapplikation

Die Demoapplikation baut auf Betriebsdaten aus der automatischen Betriebsdatenerfassung (BDE) auf (*PDA data base*). Desweiteren besteht die Möglichkeit mit Hilfe von Simulation virtuelle Betriebsdaten zu generieren (*simulation data*), um z. B. die Diagnosealgorithmen zu validieren.

Die jeweilige Datenbasis wird im nächsten Schritt mit Hilfe von SQL-Prozeduren in eine standardisierte Form (*diagnostic data*) gebracht, um von den Diagnosealgorithmen ausgewertet werden zu können. Als Hilfsmittel steht das *Gantt Tool* zur Verfügung, das die *diagnostic data* als Gantt Chart darstellt, um Fehler in der Datenbank zu erkennen oder die Diagnoseergebnisse visuell zu überprüfen. Die Struktur der Anlage wird in den *configuration & parameters* abgebildet. Dort werden die Verknüpfungen zwischen den Maschinen und zeitliche Abhängigkeiten wie Pufferkapazitäten oder Totzeiten spezifiziert. Die *batch information* dient zur sinnvollen Auswertung der Diagnoseergebnisse. Mit ihrer Hilfe können chargenbezogene Auswertungen generiert werden. Es ist aber auch denkbar anstatt der *batch information* Schichtzeiten oder Betriebszeiten zu verwenden, um entsprechende Berichte zu erzeugen.

Die eigentliche Diagnose findet in der *model based diagnosis* und/oder der *recursiv diagnosis* statt. Der Unterschied der Algorithmen besteht in der Diagnosemethode, die -wie die Namen schon sagen- zum Einen *recursiv* und zum Anderen *modellbasiert* ist. Um die Diagnoseergebnisse miteinander vergleichen zu können, wurde eine einheitliche Form der Ergebnisse gewählt (*diagnosis results*). Die Betrachtung der Resultate erfolgt mit einer Visualisierungssoftware, die die *diagnosis result* aufbereitet und als Paretochart oder Kuchendiagramm darstellt. Dabei sollten die *batch information* berücksichtigt werden, um aussagekräftige Ergebnisberichte zu erzeugen.

2 Datenbasis BDE/Simulation

2.1 PDA Data Base

Die *PDA data base* ist eine BDE-Hersteller-spezifische Datenbank. Die Datenpunkte der Maschinen sollten nach den Vorgaben der „Weihenstephaner Standards für die Betriebsdatenerfassung bei Getränkeabfüllanlagen, Version 2005“ (WS2005) aufgebaut sein. Dadurch ist sichergestellt, dass alle benötigten Betriebsdatenpunkte für die Anlagendiagnose zur Verfügung stehen. Da die Struktur der Datenarchivierung in den WS 2005 nicht festgelegt wird, ist es nötig die Datenbank in eine standardisierte Form zu bringen (*diagnostic data*), was z. B. mit SQL-Prozeduren vollzogen werden kann.

2.2 Simulation Data

Die *simulation data* ist ebenfalls eine Datenbasis, die für die Diagnose verwendet werden kann. Im Gegensatz zur *PDA data base* wird bei ihr auf Simulationsergebnisse zurückgegriffen. Die *simulation data* diente historisch zur Validierung der Diagnosealgorithmen und könnte in Zukunft verwendet werden, um bauliche Veränderungen in einer Anlage mit Hilfe einer Simulation abzusichern und die Verbesserungen des Anlagenwirkungsgrads oder den verbesserten Einfluss des optimierten Aggregates auf den Füller darstellen zu können.

3 Diagnostic Data Base

In den drei folgenden Modulen sind alle Informationen enthalten, um den Diagnosealgorithmen und der Visualisierungssoftware Berichte zu erzeugen, die Aussagen über den Einfluss von einzelnen Aggregaten auf die Füllerleistung zulassen.

3.1 configuration & parameter

In der *configuration & parameter* wird die Struktur der Anlage (d. h. die Verbindungen zwischen den Aggregaten an Hand des Materialstroms) und Parameter, die diese Verbindungen charakterisieren (z. B. Pufferkapazitäten oder Totzeiten) beschrieben. Sie sind in Form einer Excel 97-2003-Datei (*.xls)- in der sogenannten *systemdesign.xls*- abgelegt. Dazu sind zwei Tabellen von Interesse. Abbildung 2 zeigt das Tabellenblatt *mashines* in dem alle Maschinen, die sich in der Anlage befinden, beschrieben werden. Hier werden Einträge für die Aggregate, die vom Diagnosealgorithmus gesondert behandelt werden müssen, hinterlegt. So muss z. B. das Zentralaggregat gekennzeichnet sein, da die Diagnose den Einfluss der anderen Maschinen auf dieses Aggregat als Ergebnis liefern soll.

	A	B	C	D	E	F	G	H
1	mach_ID	description	central_assembly	part_of_block	td_block	t_launch_block	Kp_block	block_leader
2	1001	Depalletizer	0	0				0
3	3001	Crate Unpacker	0	0				0
4	4001	Descrewer	0	1				1
5	7001	Bottle Washer	0	0				0
6	8001	Empty Bottle Inspector	0	2	0		0	0
7	9001	Bottle Filler	1	2				1
8	10001	Labeling Machine	0	0				0
9	11001	Crate Packer	0	0				0
10	14001	Palletizer	0	0				0
11	26001	manual Sorting	0	1	41		51	0

Abbildung 2: Systemdesign.xls -> mashines

Als Zweites ist das Tabellenblatt *connection* von Interesse (siehe Abbildung 3). Hier werden die Verbindungen der einzelnen Maschinen untereinander beschrieben und Informationen über Totzeiten, Pufferkapazitäten und Materialflüsse, ob es sich beispielsweise um einen Haupt- oder Nebenstrom handelt, gegeben.

	A	B	C	D	E	F	G	H	I	J
1	source	target	Kp	td	t_launch	buffer_fillfactor	main_source	main_target	MDS_source	MDS_target
2	1001	14001	28	38	0		1	1	78	69
3	1001	4001	91	58	4		0	1	3120	3120
4	4001	3001	21	13	5		1	1	3120	3466
5	3001	11001	339	268	28		1	1	3466	2620
6	11001	14001	133	115	19		1	0	2620	2760
7	3001	7001	5200	124	28		0	1	69000	60000
8	7001	9001	3036	95	20		1	1	60000	52000
9	9001	10001	3590	120	87		1	1	50000	52000
10	10001	11001	3080	101	65		1	0	52000	52400

Abbildung 3: Systemdesign.xls -> connection

Auf eine genaue Beschreibung der Tabellen und die Bestimmung ihrer Parameter soll an dieser Stelle verzichtet werden. Bis zum Abschluss des LineMod-Projekts waren folgende Daten für die Diagnose relevant:

- Datenblatt mashines:

Spalte	Beschreibung
mach_id	Maschinen Identifikationsnummer
Description	Maschinename
Central_assembly	0 = nicht Zentralaggregat; 1 = Zentralaggregat
Part_of_block	Blocknummer bei verblockten Maschinen (nötig, wenn für Aggregate keine Daten vorhanden oder bei Verblockung)
td_block	Totzeit des Blocks
t_launch_block	Zeit bis zum Wiederanlauf nach einem Stau, wenn die Folgemaschine wieder arbeitet (historisch wird nicht mehr genutzt)
Kp_block	Max. Anzahl der Behälter im Block
Block_leader	Aggregat, das für den Block relevante Daten schreibt

- Datenblatt connection:

Spalte	Beschreibung
source	Start-Maschine der Verbindung
target	Ziel-Maschine der Verbindung
Kp	Pufferkapazität in Anzahl des zu befördernden Gebindes
td	!Achtung!: historische Bezeichnung! Hier: Summe der Totzeit und der Wiederanlaufzeit bei leerlaufenden Maschinen
t_launch	Zeit bis zum Wiederanlauf nach einem Stau, wenn die Folgemaschine wieder arbeitet (historisch wird nicht mehr genutzt)
Buffer_fillfactor	Optimaler Pufferfüllungsgrad (historisch wird nicht mehr genutzt)
main_source	1= Hauptstrom der „Quellenmaschine“, 0=Nebenstrom
main_target	1= Hauptstrom der „Zielmaschine“, 0=Nebenstrom
MDS_source	Einstellausbringung der „Quellmaschine“
MDS_target	Einstellausbringung der „Zielmaschine“

3.2 diagnostic data

Die *diagnostic data* stellt die Datenbasis der Diagnosealgorithmen da. In ihr sind intervallweise Betriebszustände einzelner Maschinen hinterlegt. Die *diagnostic data* ist aber auch ohne Probleme für die Aufnahme anderer Datenpunkte wie Counterwerte, Ausbringungen oder Bändergeschwindigkeiten geeignet. Diese Werte werden jedoch in der jetzigen Ausprägung der Diagnosealgorithmen nicht benötigt. Einen Ausschnitt aus der *diagnostic data* zeigt Abbildung 4:

mach_id	type_id	start_time	end_time	int_value	real_value
4001	2	1213160401	1213160483	128	NULL
9001	2	1213160411	1213160479	128	NULL
8001	2	1213160412	1213160484	128	NULL
3001	2	1213160427	1213160498	128	NULL
11001	2	1213160435	1213160454	128	NULL
7001	2	1213160436	1213160440	4	NULL
1001	2	1213160436	1213160458	4096	NULL
7001	2	1213160440	1213160598	128	NULL
11001	2	1213160454	1213160478	32	NULL
1001	2	1213160458	1213160459	1024	NULL

Abbildung 4: diagnostic data

Dabei beschreiben die einzelnen Spalten folgende Punkte:

Spalte	Beschreibung
mach_id	Maschinen Identifikationsnummer
type_id	Typ des Datenbankeintrags: 2 = Betriebszustand (nach WS2005)
start_time	Startzeit als unix-Zeit
end_time	Endzeit als unix-Zeit
int_value	Wert des Betriebszustands (nach WS2005)
real_value	Nur besetzt, wenn der Typ des Datenbankeintrags einen Realwert fordert

3.3 batch information

Über die *batch information* werden benötigt, um chargenabhängige Berichte zu erzeugen. An ihre Stelle können aber auch Schicht oder Betriebszeiten stehen, um entsprechende Berichte z. B. eine Schichtauswertung generieren zu können.

Im Forschungsprojekt wurden diese Daten nicht genauer betrachtet, da sie nicht im Fokus des Forschungsvorhabens lagen. Für die Zukunft müssen die Diagnoseergebnisse über die Zeit mit den *batch information* verknüpft werden, um chargenabhängige Berichte erstellen zu können. Dies geschieht in der derzeitigen Realisierung des Visualisierungstool noch per Handeingabe vor der Diagnose.

4 Gantt Tool

Das Gantt Tool ermöglicht es dem Benutzer die Datenbasis (*diagnostic data*) visuell darzustellen. Dabei werden die unterschiedlichen Betriebszustände, die über eine gewisse Zeit an den Aggregaten anliegen als Gantt Chart angezeigt (siehe Abbildung 5). So kann der Benutzer des Gantt Charts manuell auf den Verursacher eines Füllerstillstands schließen bzw. überprüfen, ob die Betriebszustände einzelner Maschinen plausibel erscheinen.

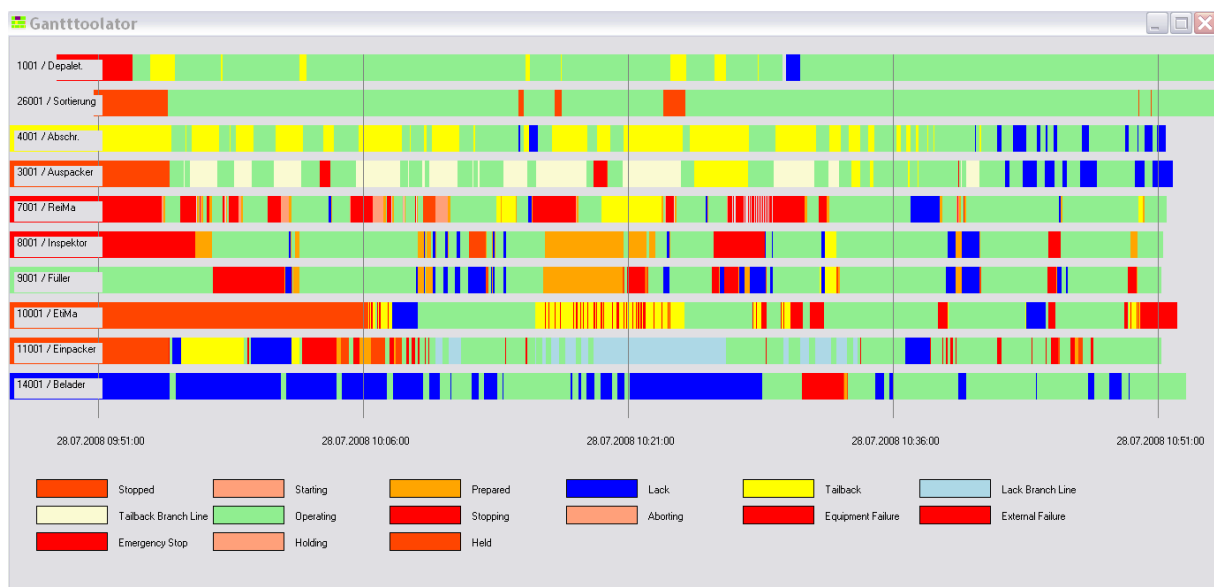


Abbildung 5: Gantt Chart

5 Diagnosealgorithmen

Für die Diagnose der Betriebsdaten stehen zurzeit zwei Algorithmen zur Verfügung. Zum Einen ein recursiver und zum Anderen ein modellbasierter. Beide Algorithmen benötigen als Input die *diagnostic data* und die *configuration & parameter*. Die Ergebnisse der Diagnose werden jeweils in die standardisierte Form der *diagnosis results* gebracht, die dann von Auswertetools graphisch verwertet werden können.

5.1 model based diagnosis

Der modellbasierte Diagnosealgorithmus wurde von der TU München, Lehrstuhl für Informatik IX, Arbeitsgruppe für Modellbasierte Systeme und Quantitatives Schließen (MQM) entwickelt. Der Ansatz vergleicht das Systemverhalten von Fehlermodellen mit dem realen Systemverhalten, auf das die Betriebsdaten schließen lassen, und findet bei der Übereinstimmung von Fehlermodell und realen Gegebenheiten die Ursache einer Störung. Durch die kontextfreie Modellierung der Anlage ist es möglich, durch Generierung neuer Fehlermodelle die Diagnoseobjekte und -mechanismen mit wenig Aufwand zu verändern. So könnte das Tool auch zur Konsistenzprüfung von BDE-Daten verwendet, neue Daten wie z.B. Ausbringungen zu einer verbesserten Diagnosegenauigkeit eingebunden oder andere Symptome wie z.B. Minderleistungen diagnostiziert werden.

5.2 recursiv diagnosis

Der recursive Algorithmus ist von der Theorie leichter aufgebaut und ist deshalb für Erweiterungen nicht so flexibel, wie der modellbasierte Algorithmus. Für die Auswertung auf Stillstandszeiten des Zentralaggregats ist er aber gut geeignet und hinreichend genau. Er wurde von der TU München, Lehrstuhl für Lebensmittelverpackungstechnik (LVT) entwickelt. Der Algorithmus geht vom Zentralaggregat aus und untersucht seine Betriebsdaten bis eine Stillstandszeit verzeichnet ist. Mit Hilfe des Betriebszustands und des darauf aufbauenden Entscheidungsbaums erkennt der Algorithmus, ob das Aggregat wegen Eigenstörung still stand oder ob das vorgeschaltete bzw. nachgeschaltete Aggregat untersucht werden muss. Beim entsprechenden Aggregat wird der gleiche Entscheidungsbaum recursiv aufgerufen bis der Stillstandsverursacher gefunden wird.

6 diagnosis result

Die Ergebnisse, die die Diagnosealgorithmen liefern werden in die *diagnosis result* geschrieben. Die *diagnosis result* ist eine xml-Datei, die nach dem folgenden Schema aufgebaut ist (Abbildung 6):

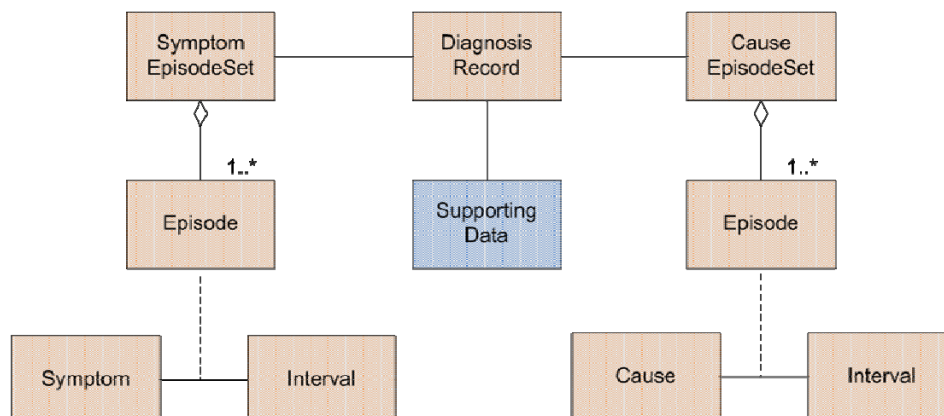


Abbildung 6: xml-Schema diagnosis result

Dabei sind die Symptome (*Symptom*) Stillstandszeiten des Füllers und die Gründe (*Cause*) die Stillstandszeiten der Symptom-verursachenden Aggregate. In den *Supporting Data* kann der Störverlauf vom Grund zum Symptom zurückverfolgt werden. So könnte beispielsweise ein Füllzustand, der durch den Auspacker verursacht wird, über die Mangelfortpflanzung an der Flaschenreinigungsmaschine beobachtet werden.

7 demo visualisation

Das Frontend der Demoapplikation (*demo visualisation*) wertet die *diagnosis results* aus und bereitet sie graphisch auf. Es wurde mit Excel 2007 und VBA umgesetzt und nutzt damit die weitreichenden Möglichkeiten von Excel, die eine einfache Generierung von Diagrammen erlauben. Abbildung 7 zeigt ein Beispiel für eine Auswertung:

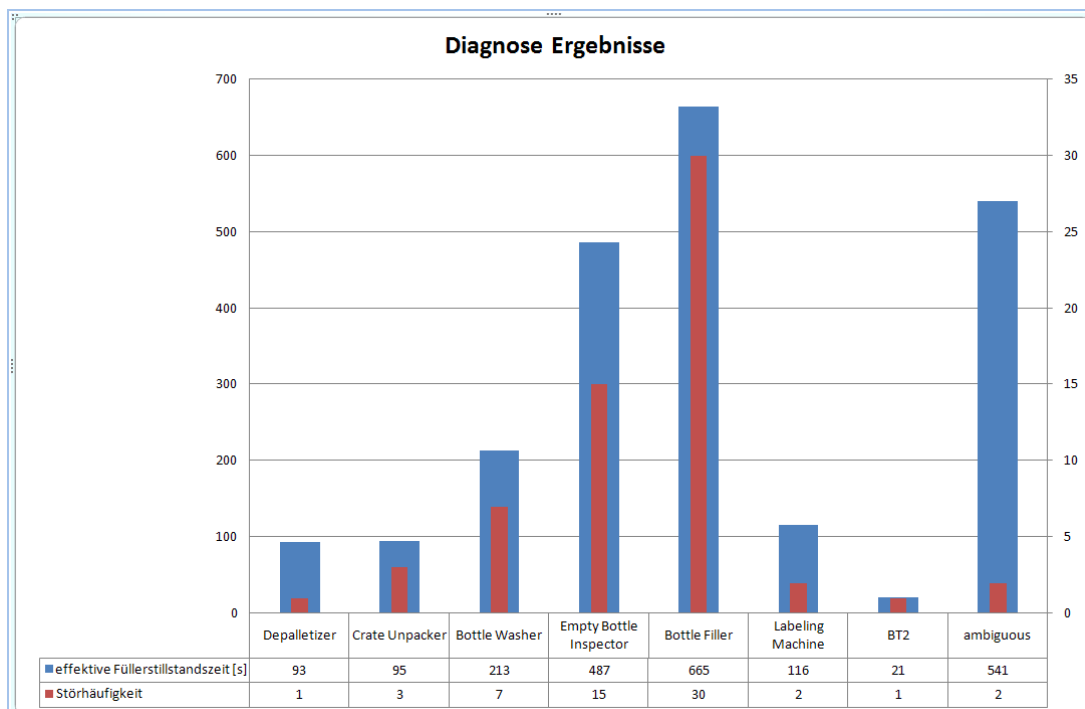


Abbildung 7: Ergebnisdarstellung Anlagendiagnose

Die aktuelle Version erlaubt eine Auswertung einer kompletten *diagnosis result*. Der Analysezeitraum wird in diesem Fall durch die Zeitauswahl beim Starten der Diagnosealgorithmen vorgegeben. Zukünftig sollten die Diagnoseergebnisse (*diagnosis result*) in einer Datenbank archiviert werden und mit Hilfe des Frontends Zeitbezüge zu den *batch informations* hergestellt werden. In der Demoapplikation ist es jedoch sinnvoller die Zeiteinstellungen beim Start des Diagnosetools zu erlauben, da so nicht immer die komplette *diagnostic data* ausgewertet werden musste, die Diagnoseergebnisse sehr leicht mittels *Gantt Tool* überprüft werden konnten und eine schnelle Reproduktion von Diagnoseergebnissen bei Veränderung der Algorithmen erzeugt werden konnte.