

Institut für Parallele und Verteilte Systeme

Abteilung Anwendersoftware

Universität Stuttgart
Universitätsstraße 38
D - 70569 Stuttgart

Studienarbeit Nr. 2345

Attributbasierte Ressourcenzuweisung in Geschäftsprozessen

Alexandru-Tiberiu Pavel

Studiengang	:	Informatik
Prüfer	:	Bernhard Mitschang
Betreuer	:	Florian Niedermann
begonnen am	:	19.01.2011
beendet am	:	22.07.2011

Abstract

In Business Processes, activities are frequently executed using both physical (e.g., machines, employees) or virtual resources. Most resource allocation procedures in BPMS choose randomly a resource which satisfies a given role. In a company a role can be occupied by more than one resource, resulting that a selection-algorithm must be used to decide which resource executes a given activity. Choosing the optimal resource is not always easy and the best possible selection can be achieved if all resource attributes like education, training data and past execution data are factored in. If not all of the data is factored in, the probability of not finding the optimal resource is increasing and suboptimal results are achieved.

This student research project presents an approach for the optimization of resource allocation in business processes. It explores both the concepts and their prototypical implementation. Business Processes are composed of business activities which can be executed parallel or sequential. They receive input data and turn it into output data which should bring benefit to the customer, supplier or the company itself. Input data in the student research project are resources. Resources are implemented as staff and machines.

The optimal resource-allocation of staff or machines along multiple dimensions like time, costs or success rates can positive influence business processes.

This student research project implements a resource manager which finds the best available resource, under consideration of several optimization parameters, in a company. To achieve this goal, several resource allocation procedures are analysed and a model-based procedure is presented which builds regression trees from the resource attributes. The regression trees can predict which resource will perform best a given activity. One algorithm which can build regression trees and is also being used in the student research project is the M5P algorithm. The M5P algorithm has the advantage over the normal regression being able to handle non linear data.

On the basis of a typical business process where resources are freely distributed, several optimization scenarios are analysed and compared how the model-based algorithm behaves compared to the standard algorithm in BPMS. At the end, expandability possibilities are discussed which can further improve the quality and the effectiveness of the resource manager.

Zusammenfassung

In Geschäftsprozesse werden Geschäftsaktivitäten oft von physischen (z.B. Maschinen oder Mitarbeitern) oder virtuellen Ressourcen ausgeführt. Die meisten Ansätze von Ressourcenallokationsverfahren in BPMS wählen zufällig eine Ressource aus, die Kriterien wie eine vorgegebene Rolle erfüllen. In ein Unternehmen kann aber eine bestimmte Rolle von mehreren Mitarbeitern erfüllt werden sodass ein Auswahlalgorithmus entscheiden muss welcher Mitarbeiter die gewünschte Geschäftsaktivität ausführen soll. Die optimale Wahl einer Ressource für eine bestimmte Geschäftsaktivität ist nicht immer leicht und um bestmögliche Ergebnisse zu erzielen müssen alle Erfolgs-wichtige Attribute wie Ausbildungs- oder Trainingsdaten sowie vergangenheitsbezogene Ausführungsdaten mitberücksichtigt werden. Werden nicht alle Daten mitberücksichtigt ist die Wahrscheinlichkeit die optimale Ressource nicht zu finden groß und es entstehen dabei suboptimale Lösungen.

Diese Studienarbeit stellt ein Ansatz zur Optimierung der Ressourcenallokation in Geschäftsprozessen vor. Es untersucht sowohl das Konzept hinter diesem Ansatz sowie die Implementierung des Ansatzes. Geschäftsprozesse bestehen aus einer oder mehreren Geschäftsaktivitäten die parallel oder sequentiell ausgeführt werden können. Sie bekommen Inputdaten und erzeugen hoffentlich Outputdaten die für Kunden, Lieferanten oder das Unternehmen selbst ein Nutzen bringt. Inputdaten die auch in der Studienarbeit verwendet werden sind Ressourcen. Implementiert sind Ressourcen vom Typ Mitarbeiter und Maschine.

Die optimale Ressourcenallokation von z.B. Mitarbeiter oder Maschinen kann Geschäftsprozesse entlang mehrerer Optimierungsdimensionen wie Zeit, Kosten oder Erfolgsquote positiv beeinflussen.

Diese Studienarbeit beschäftigt sich damit einem Ressourcen-Manager zu entwickeln, der die beste verfügbare Ressource in ein Unternehmen, unter Berücksichtigung mehrerer Optimierungsparameter, finden soll. Dazu werden mehrere Ressourcenallokationsverfahren analysiert und es wird ein Modell-basiertes Verfahren vorgestellt, welches unter Berücksichtigung der Ressourcenattribute, Regressionsbäume erstellt, die Voraussagen können welche Ressource die gewünschte Geschäftsaktivität am schnellsten ausführen würde. Ein Algorithmus welcher Regressionsbäume erstellt und auch in der Studienarbeit verwendet wird, ist der M5P Algorithmus. Der M5P Algorithmus hat den Vorteil gegenüber der Linearen Regression auch mit nicht linearen Daten umgehen zu können.

Anhand eines typischen Geschäftsprozessbeispiels wo Ressourcen frei wählbar sind, werden mehrere Optimierungsszenarien getestet und analysiert wie sich der Modell-basierte Ansatz gegenüber den Standard Ansätzen in BPMS Systemen verhält. Zuletzt werden noch Erweiterungsmöglichkeiten vorgestellt die zukünftig die Qualität und Effektivität des Ressourcen-Manager verbessern könnten.

Inhaltsverzeichnis

1	Einführung	1
1.1	Motivation.....	1
1.2	Ressourcen in Geschäftsprozesse.....	2
1.3	Struktur der Arbeit.....	4
2	Grundlagen und verwandte Arbeiten	5
2.1	dPOB-Plattform.....	5
2.2	Regressionsbäume und der MSP Algorithmus.....	6
2.3	Verwandte Arbeiten.....	11
3	Ressourcenallokation	12
3.1	Direkte Zuweisung.....	12
3.2	Statische Rollen-basierte Zuweisung.....	12
3.3	Dynamische Rollen-basierte Zuweisung.....	14
4	Implementierung	16
4.1	WEKA.....	16
4.2	JDOM.....	17
4.3	Webservices/Interfaces.....	18
4.4	Umsetzung im Websphere Kontext.....	19
5	Der Ressourcen Manager und die Satelliten Komponenten	21
5.1	Ressource-Manager Datenbank.....	21
5.2	Administrationskomponente.....	22
5.3	Ressourcen-Manager.....	25
5.3.1	Request.....	26
5.3.1.1	Parameter.....	26
5.3.1.2	Request-Verlauf.....	27
5.3.2	Release.....	29
5.3.2.1	Parameter.....	29
5.3.2.2	Release-Verlauf.....	30
5.4	Statistik Metriken.....	31
5.4.1	Ressourcenbezogene Metriken.....	31
5.4.2	Aktivitätsbezogene Metriken.....	32
5.4.3	Statistic Tabelle.....	33
6	Evaluation	34
6.1	Evaluation Entwurf.....	34
6.2	Ergebnisse.....	35
7	Abschluss und Ausblick	36
7.1	Diskussion möglicher Weiterentwicklungen.....	36
7.2	Zusammenfassung.....	37

Abkürzungsverzeichniss

API	:	A pplication P rogramming I nterface
BPMS	:	B usiness P rocess M anagement S ystemen
CDATA:		C haracter D ata
dBOP	:	d eep B usiness O ptimization P latform
DOM	:	D ocument O bject M odel
JDBC	:	J ava D ata B ase C onnectivity
JDOM	:	J ava D ocument O bject M odel
RM	:	R essourcen- M anager
SAX	:	S imple A PI for X ML
SDR	:	S tandard D eviation R eduction
SOA	:	S ervice- O riented A rchitecture
SOAP	:	S imple O bject A ccess P rotocol
WEKA	:	W aikato E nvironment for K nowledge A nalysis
WPS	:	W ebSphere P rocess S erver
WSDL	:	W eb S ervices D escription L anguage
XML	:	E xtensible M arkup L anguage

Abbildungsverzeichnis

Abbildung 1 : Standard Ressourcen Zuweisung.....	01
Abbildung 2 : Ressourcenmodel.....	02
Abbildung 3 : Rollenhierarchie.....	02
Abbildung 4 : dBOP Platform.....	05
Abbildung 5 : Regressionsbaum.....	10
Abbildung 6 : Direkte Zuweisung.....	12
Abbildung 7 : Statische Zuweisung.....	13
Abbildung 8 : Dynamische Zuweisung.....	14
Abbildung 9 : Implementierungszusammenhänge.....	16
Abbildung 10 : Webservice Beispiel.....	18
Abbildung 11 : Ressourcen Manager Schnittstelle in WID.....	19
Abbildung 12 : Implementierung in Integration Developer.....	20
Abbildung 13 : RM Datenbank.....	22
Abbildung 14 : Schritte zum Aufbau der Modelle.....	23
Abbildung 15 : Ressourcen Manager Verlauf.....	25
Abbildung 16 : Request Verlauf.....	28
Abbildung 17 : Release Verlauf.....	31
Abbildung 18 : Geschäftsprozessbeispiel.....	34
Abbildung 19 : Evaluationsergebnisse.....	35
Abbildung 20 : Dimensionen.....	37

Tabellenverzeichnis

Tabelle 1 : Ausschnitt Ressourcen-Manager – Mitarbeiter Tabelle.....	09
Tabelle 2 : Mitarbeitertabelle aufgeteilt nach den besten Split ($\leq 22,5$).....	10
Tabelle 3 : Mitarbeitertabelle aufgeteilt nach den besten Split ($> 22,5$).....	10
Tabelle 4 : Vergleich Statische und Dynamische Methode.....	16
Tabelle 5 : Welche Ausbildung hat welcher Mitarbeiter.....	25
Tabelle 6 : Ausbildungsbeschreibung.....	25
Tabelle 7 : Aus Tabelle 5 und 6 resultierende Mitarbeitertabelle.....	25
Tabelle 8 : Tabelle 7 plus Workitems.....	26
Tabelle 9 : Parameter Request Funktion.....	28
Tabelle 10 : Parameter Request Funktion.....	31
Tabelle 11 : Parameter / Beschreibung Ressourcenbezogene Metriken.....	34
Tabelle 12 : Parameter / Beschreibung Ressourcenbezogene Metriken.....	35

1. Einführung

Ein wichtiger Erfolgsfaktor für Unternehmen ist es Geschäftsprozesse effizient und effektiv auszuführen. Dadurch können Kosten gesenkt werden, Vorteile gegenüber Konkurrenz geschaffen werden und Kunden oder Lieferanten schneller bedient werden. Dies kann zur Steigerung des Umsatzes und Zufriedenheit der Kunden führen.

Ein Geschäftsprozess kann mehrere Geschäftsaktivitäten sequentiell oder parallel ausführen um ein gewünschtes Ergebnis zu erzielen. Eine Aktivität wird von einer oder mehreren Ressourcen ausgeführt. Allgemein gesagt sind Ressourcen materielle oder immaterielle Entitäten. In dieser Arbeit sind Ressourcen Mitarbeiter oder Maschinen eines Unternehmens. Mitarbeiter und Maschinen besitzen mehrere Attribute, welche Erfolgs-wichtige Eigenschaften, wie Ausbildungs-, Trainingsdaten sowie Maschinen-typische Merkmale enthalten.

Wie schnell, kostspielig oder erfolgreich eine Aktivität ausgeführt wird hängt stark von der gewählten Ressource ab. Dem Ressourcen-Auswahlprozess kommt deshalb eine herausragende Bedeutung zu. Diese Studienarbeit beschäftigt sich mit der Implementierung eines Ressourcen-Managers der basierend auf die Attribute der Ressource, Aktivitätsspezifische Regression-Modelle aufbaut um voraussagen zu können welche Ressource am schnellsten die gewünschte Aktivität ausführen kann [19, 21].

Im ersten Unterkapitel wird der Standard Ansatz von Ressourcenzuweisungen erläutert und es wird erklärt welche Probleme solch ein Ansatz verbirgt. In Unterkapitel 1.2 wird definiert was Ressourcen im Unternehmen sind und welche Ressourcen in der Studienarbeit verwendet werden. Letztens werden vergleiche mit Verwandte Arbeiten gemacht.

1.1 Probleme/Motivation

Viele Geschäftsprozesse werden mit Hilfe von Business Process Management Systemen (BPMS) beschrieben [28]. Diese besitzen Integrierte Lösungen zur Ressourcen Auswahl und die meisten Ansätze folgen dem in Abbildung 1 beschriebenen Verlauf.

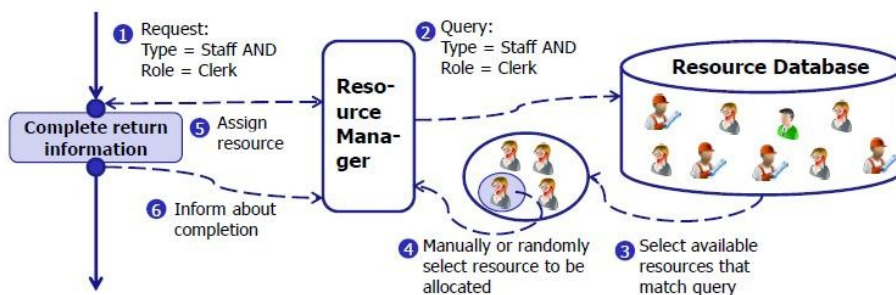


Abbildung 1: Standard Ressourcen Zuweisung [1]

Wenn eine Ressource-Anfrage gestartet wird, sucht der Ressourcen-Manager die Datenbank nach allen verfügbaren Ressourcen, welche die Anfrage Bedingungen erfüllen. Die meisten Anfragen suchen nach einem bestimmten Typ wie Maschine oder Mitarbeiter und die formale Qualifikation (Rolle) einer Maschine oder eines Mitarbeiters in der Unternehmenshierarchie.

Wenn mehr als eine Ressource gefunden wird, wählt der RM zufällig eine Ressource aus. Die Aktivität wird mit der gewählten Ressource ausgeführt [1].

Die Herausforderung bei diesem Ansatz ist, dass die voraussichtliche Leistung der Ressource bei der Ausführung der Aktivität nicht berücksichtigt wird und so möglicherweise suboptimale Ressourcen ausgewählt werden. Es besteht daher ein Optimierungspotenzial darin, diesen Zuordnungsvorgang dahingehend umzugestalten, dass nur Ressourcen mit einer "guten" voraussichtlichen Leistung selektiert werden.

Das Beispiel aus Kapitel 6 simuliert einen typischen Versandhandel mit Verkäufern und Mechanikern. Mechaniker werden nur dann eingesetzt wenn eine Maschine einen Defekt aufweist und repariert werden muss. 25% der Angestellten sind Mechaniker und 65% sind Verkäufer. Möchte man seine Kunden so gut und schnell wie möglich über den Status des Versandes informieren, ist es ausschlaggebend, welche Ressource für die Mechanikerrolle oder Verkäuferrolle ausgewählt wird. Da eine Ressource in einem Geschäftsprozess mehrere Aktivitäten sequentiell ausführen kann, wie es auch im Falle des Beispiels aus Kapitel 6 vorkommt, werden zusätzliche Optimierungsparameter besprochen und weiter in Kapitel 4 erläutert. Die Optimierungsmöglichkeiten analysieren ob die ausgewählte Ressource eine gewisse Aktivität optimal ausführen kann, oder lieber eine Ressource ausgewählt wird die alle Aktivitäten des Geschäftsprozesses, die für Sie bestimmt sind optimal ausführen kann.

Um optimale Ergebnisse zu erzielen muss man die Attribute der gewählten Ressourcen sowie vergangenheitsbezogene Ausführungsdaten, welche eine entscheidende Rolle in der Aktivitätsausführungszeit haben können, berücksichtigen. Ein Mitarbeiter welcher ein Mechaniker-Zertifizierungskurs abgeschlossen hat, repariert mit einer höheren Wahrscheinlichkeit schneller ein Automobil als ein Kollege der keine Kurse in Richtung Mechanik besucht hat.

Welche Ressourcenallokationsverfahren möglich sind und wie diese in einem Geschäftsprozessszenario verwendet werden wird ab Kapitel 3 erklärt.

1.2 Ressourcen in Geschäftsprozessen

Geschäftsprozesse besitzen eine oder mehrere Geschäftsaktivitäten. Eine Geschäftsaktivität wird von einer oder mehreren Ressourcen ausgeführt. Die optimale Ausführung eines Geschäftsprozesses hängt somit stark von den eingesetzten Ressourcen ab.

Je nach Prozess können Ressourcen unterschiedlicher Art sein. So definiert [29] Ressourcen wie folgt : „Als Ressource [...] wird im Allgemeinen ein Mittel bezeichnet, das benötigt wird, um eine bestimmte Aufgabe zu lösen. Meisten werden darunter Betriebsmittel, Geldmittel, Boden, Rohstoffe, Energie aber auch Personen verstanden“ [29].

In der Studienarbeit sind Ressourcen entweder Personen (Mitarbeiter eines Unternehmens) oder Maschinen.

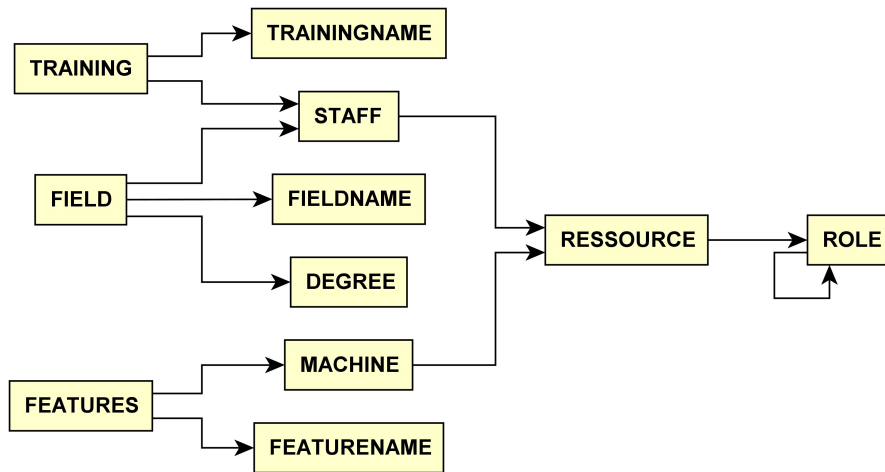


Abbildung 2: Ressourcenmodell

Jede Ressource hat eigene typische Eigenschaften (Attribute) die Sie kennzeichnet (Abbildung 2). Somit haben Mitarbeiter Ausbildungen (Field) mit einem speziellen Abschluss (Degree) oder Trainings und Maschinen haben spezielle technische Merkmale (Features). Diese Eigenschaften bestimmen wie effizient und schnell eine Ressource eine Aktivität ausführen würde.

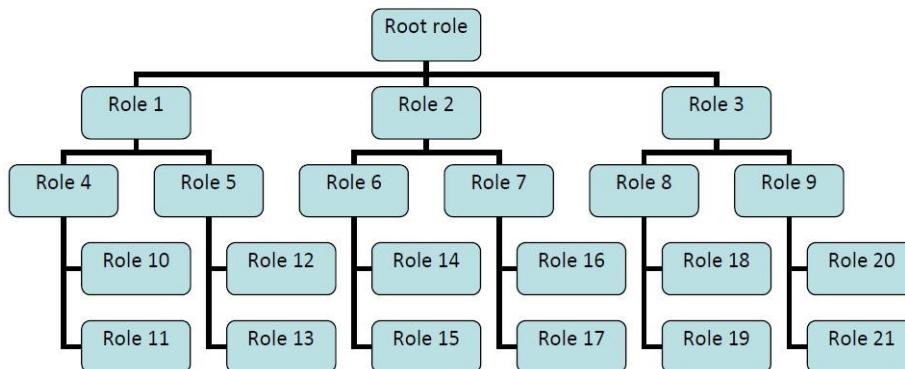


Abbildung 3: Rolenhierarchie

Jede Ressource hat eine Rolle, welche dessen formale Qualifikation für eine oder mehrere Aktivitäten festlegt. Rollen sind in eine Spezialisierung-Generalisierung Hierarchie wie in Abbildung 3 aufgebaut. Eine Rolle hat genau einen Vaterknoten und ein Vaterknoten kann mehrere Kinder haben. Wichtig zu erwähnen ist, dass eine Rolle qualifiziert ist, Aktivitäten von den Vaterknoten auszuführen. Rolle 10 könnte die Aktivitäten die für Rolle 10, Rolle 4, Rolle 1 oder Root Rolle bestimmt sind, ausführen. Eine Rolle kann jedoch nicht die Aktivitäten die für seine Geschwister gedacht sind ausführen.

1.4 Struktur der Arbeit

In Kapitel 2 werden die Grundlagen hinter den Ressourcen-Manager erklärt. Dabei wird die dBOP Plattform erklärt sowie die mathematischen Grundlagen die hinter der Linear Regression und den Regressionsbäumen bzw MSP Algorithmus versteckt sind. Der letzte Unterkapitel beschreibt Ähnlichkeiten und Unterschiede zwischen dem in dieser Studienarbeit vorgestellten Ressourcen-Manager und Verwandte Arbeiten. Kapitel 3 stellt verschiedene Ressourcenallokationsmethoden vor und beschreibt Vorteile und Nachteile des jeweiligen Verfahren. Kapitel 4 kümmert sich um alle Konzepte die notwendig sind um den Ressourcen-Manager zu implementieren. Dabei werden die Java Bibliotheken WEKA und JDOM vorgestellt, sowie die Umsetzung des Ressourcen-Managers als Webservice mit Hilfe von IBM Websphere Process Server und IBM Websphere Integration Developer. Die Implementierung des Ressourcen-Manager besteht aus 3 Komponenten. In Kapitel 5 wird die Umsetzung dieser 3 Komponenten beschrieben. Wie sich der Ressourcen-Manager in einem Geschäftsprozess mit mehrere Geschäftsaktivitäten verhält wird in Kapitel 5 studiert. Es werden mehrere Tests mit verschiedenen Einstellungen aufgebaut und die Ergebnisse werden verglichen und analysiert. Letztens, in Kapitel 7, werden mögliche Erweiterungen vorgestellt und es wird zusammengefasst was in allen Kapiteln vorgestellt wurde.

2. Grundlagen und Verwandte Arbeiten

Dieses Kapitel stellt die Hintergründe vor, auf welche die Implementierung der Studienarbeit basiert. Die deep Business Optimization Platform, welche im Unterkapitel 2.1 erläutert wird, ist die Basis hinter der Studienarbeit und der Ressourcen Manager wird als eine Komponente innerhalb dieser Platform implementiert. Als nächstes, in Unterkapitel 2.2 werden die mathematischen Grundlagen hinter den Voraussagemodell erklärt und anhand eines Beispiels verdeutlicht wie die Regressionsbäume erstellt werden. Zuletzt werden verwandte Arbeiten mit dieser Arbeit verglichen.

2.1 Deep Business Optimization Platform

Um optimale Lösungen zu bekommen benötigen wir einen Ressourcen-Manager, welcher voraussagen kann welche Ressource am besten eine Aktivität ausführen könnte. Eine Plattform die das ermöglicht ist die deep Business Optimization Platform [2]. Der Ressourcen-Manager ist Teil der deep Business Process Optimization Schicht und der Execution Komponente. Die dBOP, welche in Abbildung 3 dargestellt ist, besteht aus 3 Schichten.

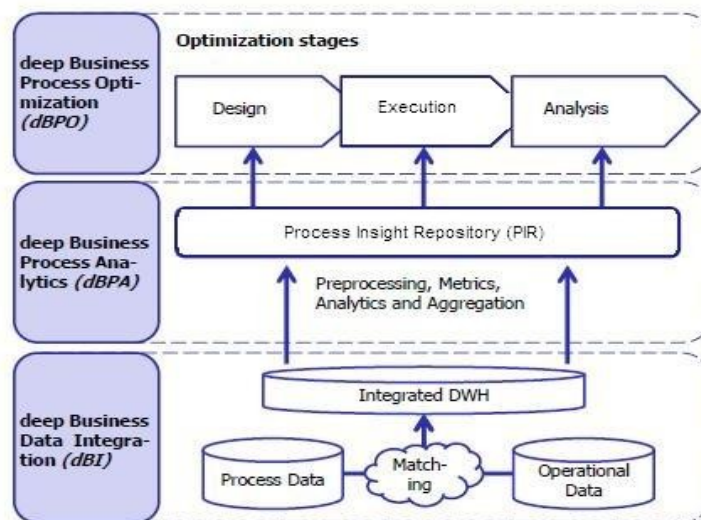


Abbildung 4: dBOP Plattform [1]

- **Data Integration**
Relevante Ressourcendaten können auf mehrere Datenquellen verteilt, in unterschiedlichen Format und heterogen sein. Die Data Integration Schicht [22] kümmert sich darum diese Daten zu bereinigen, normalisieren, standardisieren und eine globale Sicht auf diese Daten zu schaffen, damit man auf alle wichtigen Daten in der Analyseschicht zugreifen kann.

- **Analyseschicht**
Die Analyseschicht [3] ermöglicht Prozess- und Ressourcen-orientierte Metriken aus den integrierten Daten der ersten Schicht zu berechnen. Mit Hilfe der Metriken ist es dann möglich, Klassifikationsmodelle aufzubauen, die voraussagen können, welche Ressource am schnellsten eine gewünschte Aktivität ausführen kann. Die aggregierten Daten zusammen mit den Modellen werden in den Process Insight Repository gespeichert [2].
- **Optimierungsschicht**
Die Ergebnisse der Analyseschicht werden in der Optimierungsschicht benutzt, um die Optimalität des Business Process Designs und der Business Process Ausführung in Bezug auf die Zielfunktion sicherzustellen. Dafür benutzt man eine Menge sogenannter Muster. Muster sind formalisierte Optimierungstechniken [4].

2.2 Regressionsbäume und der M5P Algorithmus

In diesen Unterkapitel werden die statistischen Grundlagen hinter den Voraussagemodellen erklärt. Es existieren mehrere Klassifikationsalgorithmen, die Voraussagen können, welche Ressource eine bestimmte Aktivität am besten ausführen würde. Die Klassifikationsalgorithmen konstruieren anhand von Trainingsdaten Klassifikationsmodelle. Ein Algorithmus, der solche Modelle aufbauen kann, ist der M5P Algorithmus. In den nächsten Paragraphen wird der M5P Algorithmus erklärt. Dabei wird zuerst auf die Lineare Regression sowie auf die Multilineare Regression eingegangen, weil M5P im Effekt mehrere lineare Regressionsmodelle, die in einem Baum verteilt sind, darstellt. Danach werden Regressionsbäume und der M5P Algorithmus eingeführt.

Jede Ressourcen-typische Eigenschaft kann eine entscheidende Rolle in der Ausführungszeit einer Aktivität haben. Wie schnell ein Mitarbeiter eines Unternehmens eine Aktivität ausführt, hängt stark z.B. von dessen Ausbildung oder Training ab. Die Regressionsanalyse berechnet, wie stark eine Variable Y von anderen Variablen X_1, \dots, X_N abhängt. In dieser Studienarbeit ist Y die Ausführungszeit einer Aktivität und X_1, \dots, X_N die Ressourcen-Attributwerte.

Die Lineare Regressionsformel[14, 15] lautet :

$$Y_j = \beta_0 + \beta_1 * x_{1j} + \beta_2 * x_{2j} + \beta_3 * x_{3j} + \dots + \beta_n * x_{nj} + \varepsilon_j \quad j = 1, 2, \dots, m$$

Formel 1: LineareRegression

Wobei :

- Y** : ist die Abhängige Variable, unser Zielwert
- x_1, \dots, x_n** : sind die unabhängigen Variablen, entsprechen der Einflussgrößen
- e** : zufälliger Fehler der nicht beobachtbar ist
- β_1, \dots, β_n** : sind die (partiellen) Regressionskoeffizienten

Diese *Formel 1* kann umgeschrieben werden zu :

$$E(y_j) = \beta_0 + \sum_{i=1}^n \beta_i * x_{ij} \quad , \quad \text{da} \quad \sum_{i=1}^n e_i = 0$$

Ziel ist es den Fehler e , welcher der Summe der quadrierten Abweichungen zwischen Vorhersagewerten und Beobachtungswerten entspricht, zu minimieren.

Die Formel 1 kann als Matrixschreibweise zu folgender Formel umgeschrieben werden :

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \dots \\ Y_n \end{bmatrix} = \begin{bmatrix} \beta_0 + \beta_1 * X_1 \\ \beta_0 + \beta_2 * X_2 \\ \dots \\ \beta_n + \beta_n * X_n \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \dots \\ e_n \end{bmatrix} \quad \Leftrightarrow$$

$$\Leftrightarrow \begin{bmatrix} Y_1 \\ Y_2 \\ \dots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \dots & \dots \\ 1 & X_n \end{bmatrix} * \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \dots \\ e_n \end{bmatrix}$$

Daraus folgt das $\vec{e} = \vec{Y} - \vec{X} * \vec{\beta}$

Zur Minimierung des Fehlers minimiert die lineare Regression die quadrierte Abweichungen.

$$\sum e_i^2 = [e_1 \ e_2 \ \dots \ e_n] \begin{bmatrix} e_1 \\ e_2 \\ \dots \\ e_n \end{bmatrix} = \vec{e}^T \vec{e}$$

$$\vec{e}^T \vec{e} = (\vec{Y} - \vec{X} * \vec{\beta})^T * (\vec{Y} - \vec{X} * \vec{\beta})$$

$$\frac{d}{d\beta} (\vec{Y} - \vec{X} * \vec{\beta})^T * (\vec{Y} - \vec{X} * \vec{\beta}) = -2 * \vec{X}^T * (\vec{Y} - \vec{X} * \vec{\beta})$$

$$-2 * \vec{X}^T * (\vec{Y} - \vec{X} * \vec{\beta}) = 0 \equiv \vec{X}^T * \vec{Y} = \vec{X}^T * \vec{X} * \vec{\beta}$$

$$\rightarrow \vec{\beta} = (\vec{X}^T * \vec{X})^{-1} * \vec{X}^T * \vec{Y}$$

Die Koeffizienten β können also mit der folgenden Formel berechnet werden :

$$\vec{\beta} = (\vec{X}^T * \vec{X})^{-1} * \vec{X}^T * \vec{Y}$$

β_0 kann jetzt leicht berechnet werden indem man Werte für X in der *Formel 1* einsetzt [14].

Die Lineare Regression bildet ein globales Model, welcher für die ganze Datenmenge zuständig ist. Bei großen Datenmengen, wie bei unserer Ressourcen-Datenbank, entsteht oft ein nichtlineares Muster welcher die Voraussage leicht bis stark stören kann.

Eine Lösung für dieses Problem bietet der M5P Algorithmus, der auch als WEKA Bibliothek verfügbar ist. Ziel ist es die Datenmenge zu partitionieren, weil man somit die Daten leichter verwalten kann.

Beim M5P partitioniert man die Daten rekursiv bis alle Daten durchlaufen sind. Die Partitionen werden benutzt um Entscheidungsbäume zu erstellen. In den inneren Knoten werden die Attribute gespeichert und die Kante die zwei Knoten verbindet enthält Regeln.

Die Regeln werden schnell ausgewertet, da diese boolesche Ausdrücke der Form *Wert θ Konstante*, wobei $\theta = \{ < , > , \leq , \neq , \geq , = \}$, sind. In den Blättern stehen die Linear Regression Modelle. Welches das Split-Attribut ist, wird entschieden mit Hilfe der Standardabweichung und des Standard Deviation Reduction [13].

Zur besseren Verdeutlichung wird der M5P Anhand eines kleines Beispiels erklärt.

Alter	Rating	Informatik	Dauer
20	2	0	40
25	7	1	10
45	10	1	15
25	3	0	34
20	5	0	27
45	9	1	5
25	6	1	19
45	8	1	22

Table 1: Ausschnitt Ressourcen-Manager - Mitarbeiter Tabelle

1. Als erstes wird die Standardabweichung des Klassenattributs berechnet

$$S = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2} \quad S_{\text{DAUER}} = 12,79$$

Als nächstes muss der Begriff des Standard Deviation Reduction erklärt werden. Die Standard Deviation Reduction nimmt an das nach einen Split die Standardabweichung des Klassenattributs sinkt.

Die Formel des Standard Deviation Reduction ist :

$$SDR(T, X) = S(T) - S(T, X) \quad ; T = \text{Klassenattribut}, X = \text{Split Attribut}$$

$S(T, X) = \sum_{c \in X} P(C) * S(C)$, wobei P(C) die Wahrscheinlichkeit des Vorkommens von Attributwert X_c in den Daten entspricht und S(C) = Standardabweichung von Attribut X_c

Um zu sehen welches Attribut am informativsten ist, nimmt man das Attribut X welches S(T,X) minimiert bzw. SDR(T,X) maximiert.

Zurück zum Beispiel :

$$\begin{aligned} & S(\text{Dauer}, \text{Alter}(\text{Alter} < 22,5)) = \\ & P(\text{Rating}) * S(\text{Rating}) + P(\text{Informatik}) * S(\text{Informatik}) \\ & = \frac{2}{7} * 9,19 \quad + \quad \frac{5}{7} * 11,06 \quad = 9,21 \end{aligned}$$

$$\begin{aligned} \text{SDR}_1(\text{Dauer}, \text{Alter} (22,5)) &= 11,84 - 9,21 = 2,63 \\ \text{SDR}_2(\text{Dauer}, \text{Alter} (35)) &= 11,84 - 10,64 = 1,20 \end{aligned}$$

$$\max(\text{SDR}_1, \text{SDR}_2, = (\text{Dauer}, \text{Alter} (22,5)) = 2,63$$

Der erste Splitt geschieht mit den Attribut Alter (22,5).
Die daraus entstehende Partitionen sehen folgendermaßen aus :

Alter <= 22,5	Rating	Informatik	Dauer
20	2	0	40
20	5	0	27

Table 2: Mitarbeiter Tabelle aufgeteilt nach dem besten Split (Alter <=22,5)

Alter > 22,5	Rating	Informatik	Dauer
25	7	1	10
45	10	1	15
25	3	0	34
45	9	1	5
25	6	1	19
45	8	1	22

Table 3: Mitarbeitertabelle aufgeteilt nach dem besten Split (Alter > 22,5)

Die Daten werden dann so lange Partitioniert bis keine neuen Informationen gewonnen werden können. Also wenn alle SDR Werte von den übrigen Attributen kleiner als eine gewisse Schranke sind oder wenn nur eine bestimmte Anzahl an Daten übrig geblieben sind [23, 13].

Der resultierende Baum für das obige Beispiel wird in Abbildung 4 dargestellt und sieht folgendermaßen aus:

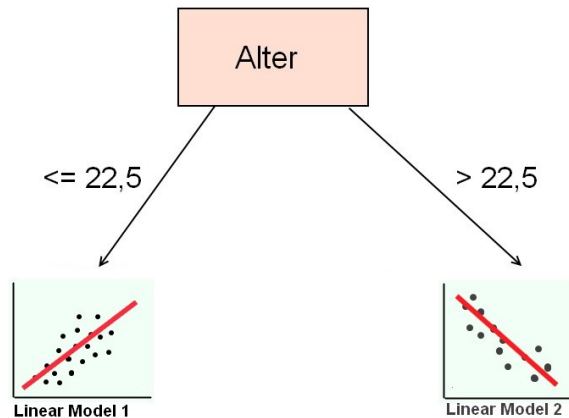


Abbildung 5: Regressionsbaum

Möchte man jetzt Voraussagen wie schnell ein Mitarbeiter eine gewisse Aktivität ausführen würde (Dauer) wenn Alter=33, Rating=9 und Informatik=1 sind, geht man folgendermaßen vor :

Die Wurzel des Baumes ist Alter, deswegen nimmt man zuerst den Wert von Alter. Man bemerkt das Alter = 33. Der Pfad führt direkt in ein Blatt Knoten. In den Blätter sind die Linear Regression Modelle gespeichert und man wendet die Formel 1 mit den gewünschten Werten.

$$Dauer = b_0 + b_1 * Alter + b_2 * Rating + b_3 * Informatik + e$$

Formel 2: LM2 aus Abbildung 4

$$Dauer = 11,36 + 0,4208 * 33 + 0 * 9 + 0 * 1 + e$$

Somit wäre die geschätzte Dauer für ein Mitarbeiter mit Alter=33, Rating=9 und Informatik=1 gleich 25,25.

2.3 Verwandte Arbeiten

Die Studienarbeit ist Teil der dBOP Plattform. Die Data Integrationsschicht ist in [3] besprochen. Die Methoden die in der Analyseschicht verwendet werden, sind Standard Mining Techniken besprochen in [5][6]. Beispielapplikationen können in [2] gefunden werden. Die Rolle innerhalb des dBOP ist Thema von [4].

Der Ansatz der dieser Studienarbeit am nächsten kommt ist der klassifizierungs-basierte Algorithmus der in [7] und [8] beschrieben wird. Unterschiede sind je-

doch das der in dieser Studienarbeit implementierte Ressourcen-Manager zwei Optimierungszielfunktionen besitzt und nicht nur eine wie in [7] und [8]. Dem in [7] und [8] vorgestellten Ansatzes fehlen Integrations-Möglichkeiten die die Qualität der Daten stark verbessern kann. Außerdem ist es unklar wie der Ressourcen-Manager in einer Prozessausführung eingesetzt wird. [7] und [8] sammeln alle Workflow Informationen aus den Event Log. Der Event Log besteht aus Tupel mit folgender Struktur : Aktivität, Mitarbeiter. Auf diese Daten werden die Klassifikatoren trainiert. Als Klassifikator wird der C4.5, Naive Bayes und Support Vector Machine Algorithmus benutzt. Die Attribute der Mitarbeiter werden in diesen Ansatz nicht mitberücksichtigt, welche eine entscheidende Rolle in der Ausführungszeit der Aktivitäten haben können. Ein weiterer Unterschied ist das in [7] und [8] das Verfahren semi-automatisch ist. Das bedeutet dass nachdem der Klassifikator ein Mitarbeiter ausgewählt hat, ein Prozessanalyst entscheiden muss, ob der gewählte Mitarbeiter tatsächlich der Beste für die gewünschte Aktivität ist.

[9] zeigt die Machbarkeit eines solchen Ansatzes in einem Software Engineering Szenario. Dabei wird versucht, Bug Reports von Benutzern an Entwickler weiterzuleiten die ähnliche Bugs erfolgreich gelöst haben. Dafür wird Text Categorization aus dem Bereich maschinelles Lernen verwendet. Dies involviert die Klassifikation von Textdokumenten in eine Menge von Kategorien [31]. Bei dem Ansatz sind die Textdokumente die Bugreports und die Kategorien sind die Namen der Entwickler.

[10] baut Mehrdimensionale Assoziationsregeln mit Hilfe des Apriori Algorithmus. Je höher der resultierende Konfidenzoeffizient desto wahrscheinlicher ist es den optimalen Mitarbeiter gefunden zu haben. Dieser statische Ansatz kann bei Mitarbeitern, zu denen keine Daten existieren, keine Ergebnisse liefern.

[11] Benutzt Mining aus dem Workflowdaten. Aus diesen Daten werden dann Entscheidungsbäume erstellt um Mitarbeiterzuweisungsregeln aus dem Ausführungsdaten des Workflows erstellen zu können. Zusätzlich wird noch untersucht, wie man die gewonnen Regeln als Inputdaten benutzen könnte um z.B die Arbeitsbelastung zu optimieren [1].

3. Ressourcen Zuweisung

In ein Geschäftsprozessszenario gibt es mehrere Möglichkeiten eine Ressource einer bestimmten Aktivität zuzuweisen [20]. Welche Ressource zugewiesen wird beeinflusst Geschäftsprozessresultatfaktoren wie Kosten, Qualität und Zeit. In den meisten BPMS Systemen werden Ressourcen folgend Charakterisiert :

- Ressourcen Rolle : *RO*
- Ressourcen Typ : *T*
- Ressourcen Attribute : *ATT*

In den nächsten Unterkapiteln werden drei Methoden zur Ressourcenallokation vorgestellt und analysiert welche Vorteile und Nachteile der jeweilige Ansatz hat. Im ersten Unterkapitel wird die direkte Zuweisung, als nächstes die statische Zuweisung und letztens die dynamische Zuweisung sowie die verschiedenen Optimierungsparameter die bei der dynamischen Zuweisung implementiert werden können erklärt.

3.1 Direkte Zuweisung

Bei der direkten Zuweisung [20] wird direkt spezifiziert, z.B. durch Eingabe der ResourceID, welche Ressource aus der RM Datenbank gewählt wird. Dies bringt Vorteile nur wenn genau eine Person für die Ausführung der gewünschten Aktivität in Frage kommt. Nachteile entstehen wenn mehrere Personen die Aktivität ausführen könnten, weil es passieren kann, dass die direkt gewählte Ressource nicht Verfügbar ist. Nicht Verfügbar, weil sie beschäftigt ist, eine andere Aktivität auszuführen oder frei genommen hat aus persönlichen oder gesundheitlichen Gründen. Eine bessere Alternative liefert die nächste vorgestellte Methode .

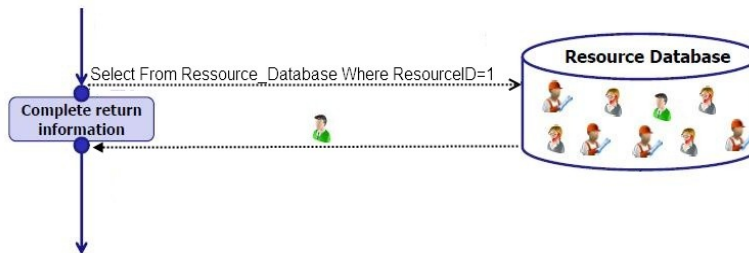


Abbildung 6: Direkte Zuweisung

3.2 Statische Rollen-basierte Zuweisung

Bei der statischen Methode [20, 1] (Abbildung 5) wählt ein Prozessanalyst die Attribute aus, die die Aktivitätsperformance positiv beeinflusst. Danach wird die Query verfeinert um sicherzustellen dass die gewählten Ressourcen nicht nur Stan-

dar Anforderungen erfüllen sondern voraussichtlich die gewünschte Aktivität auch gut durchführen werden.

Bei der statischen Query unterscheidet man zwischen zwei Abfragemöglichkeiten. Diese Ansätze haben den Vorteil, dass die Ressourcen Management Applikation nicht weiter verändert werden müsste, bringen aber auch Nachteile mit sich. Die Ansätze werden in Abbildung 7 dargestellt.

Bei der Ersten (Abbildung 7, Basic Query) wird als Parameter nur die Rolle und der Typ übergeben und danach wird zufällig eine Ressource gewählt die die gewünschte Rolle und Typ erfüllt. Dieser Ansatz ähnelt stark den Ansatz, der in Unterkapitel 3.1 beschrieben wurde und hat den Nachteil, das möglicherweise viele Ressourcen zurückgeliefert werden. Der Prozessanalyst muss dann entscheiden welche Ressource dann tatsächlich am besten für die Aktivitäten geeignet ist.

Bei der zweiten Methoden (Abbildung 7, Additional conditions) wird die zurückgelieferte Menge an Ressourcen verringert indem zusätzliche Attribute wie Erfahrungen oder Trainingsdaten als Parameter geschickt werden um die Suche zu verfeinern. Der Nachteil ist dass es manchmal sehr schwierig ist manuell eine optimale Query aufzubauen. Diese Queries sind dann sehr komplex und erfordern viel Zeit und Aufwand bei Änderungen. Es kann auch zu Verschlechterungen der Performance führen weil die Anzahl der möglichen Ressourcen durch diese komplexe Queries verringert werden, was zu Verzögerungen oder sogar Engpässen führen kann.

Eine Typische Statische Query wird folgend definiert :

- $T \times Ro \times Rel[x f(AC)] \rightarrow R$ wo :
- $T=Typ, Ro=Rolle$ und $R=Ressource$ (Ansatz 1)
- $AC \in Att \times Comp \times Domain(Att)$ (Ansatz 2), beschreibt die Eigenschaften welche die Ressourcen Attribute erfüllen müssen, mit $Comp \in \{=, <, >, = >, = <, \in\}$ und $f(AC)$ beschreibt alle Eigenschaftsvergleichsmöglichkeiten.

Der Typische Verlauf einer Statischen Abfrage wird in Abbildung 6 dargestellt.

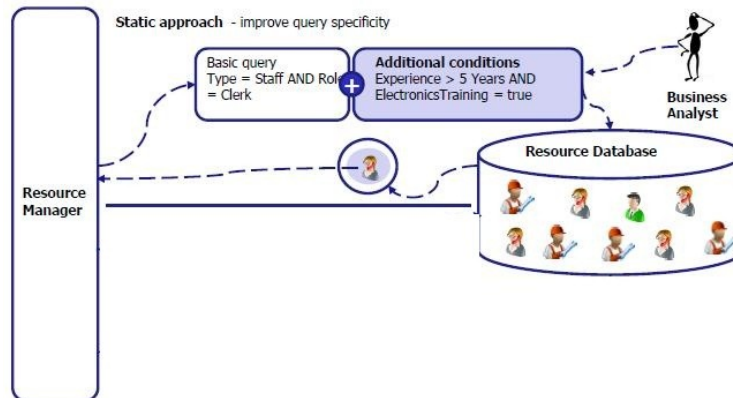


Abbildung 7: Statische Methode [1]

Eine verbesserte Variante wäre die dynamisch Rollen-basierte Zuweisung, welche in den nächsten Unterkapitel besprochen wird.

3.3 Dynamisch Rollen-basierte Zuweisung

Bei der Dynamischen Zuweisung [1] wird anstatt der komplexen Queries ein Voraussagemodell konstruiert. Solch ein Ansatz erfordert die Veränderung des Ressourcen Managers und die Einführung einer Ausführungsdatentabelle. Trotz der notwendigen Veränderungen erzielt der dynamische Ansatz bessere Ergebnisse gegenüber dem statischen Ansatz. Vorteilhaft ist das keine manuellen Veränderungen durchgeführt werden müssen. Ein weiterer Vorteil ist die Flexibilität des Ansatzes. Beliebig viele Daten können hinzugefügt werden um zu analysieren wie diese die Aktivitätsausführung beeinflussen. Flaschenhals-Situationen entstehen bei den dynamischen Ansatz viel seltener als bei einen statischen Ansatz.

Der Verlauf eines Requests in einen dynamischen Ansatz ist in Abbildung 7 zu sehen.

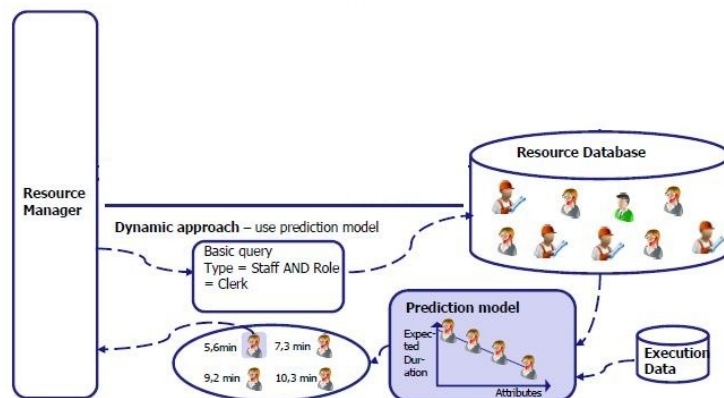


Abbildung 8: Dynamische Methode [1]

Der Aufbau der Modelle kann auf mehrere Dimensionen optimiert werden. Zwei Optimierungen welche auch im Ressourcen-Manager implementiert worden sind, sind die Zielfunktionen (GoalFunctions).

Die erste Zielfunktion „Current_Activity“ analysiert wie gut eine Ressource eine gewisse Aktivität ausgeführt hat. Dafür werden alle Ausführungsdaten der gewünschten Aktivität in eine Tabelle mit den Namen der Aktivität gespeichert. Auf diese Daten werden dann die Klassifikatoren trainiert.

Die zweite Zielfunktion „All_Activities“ analysiert, wie gut eine Ressource alle Aktivitäten, in welche sie involviert war, ausgeführt hat. Es kann sein, dass eine Ressource Aktivität A gut ausgeführt hat und Aktivität B etwas schlechter. Ziel dabei ist es, die beste Ressource zu finden die alle Aktivitäten am besten ausgeführt hat.

Eine weitere Optimierung kann das Anfügen von Workitems sein. Dieser Ansatz wird auch in der Studienarbeit verfolgt. Jede Aktivität kann zusätzliche Eigenschaften (Workitems) besitzen, welche die Modelle beeinflussen können. Workitems können vom Typ String, Numerisch oder Boolean sein und definieren zusätzliche Merkmale, welche eine Ressource zur erfolgreichen Ausführung der Aktivität besitzen sollte.

Weitere Optimierungen können auf der Ebene des Optimierungszieles sein. Optimierungsziele können *Dauer* der Ausführung, *Kosten* die eine Ressource zur Ausführung der Aktivität beansprucht, *Flexibilität* einer Ressource oder *Erfolgsquote* sein. In der Studienarbeit wurde als Optimierungsziel die Dauer gewählt. Die weiteren Optimierungsziele werden in Kapitel 7.1 erklärt.

Die untere Tabelle stellt die statische Methode gegenüber der dynamische und fasst alle Vorteile und Nachteile zusammen.

Statischen Methode	Dynamische Methode
+ Keine Änderung des Ressourcen-Managers notwendig	+ Ermöglicht die Aufnahme von komplexen Kriterien bei der Auswahl
+ Erlaubt manuelle Anpassungen	+ Abfragekomplexität wird nicht erhöht
- Die Komplexität der Auswahlkriterien ist eingeschränkt	+ Anpassung benötigt nur das Wiedertrainieren des Voraussagemodells
- Komplexe Abfragen möglich	+ Kein erhöhtes Risiko von Engpässen (Flaschenhals)
- Zukünftige Anpassungen schwierig und erfordern große Anstrengungen	- Erfordert Änderungen an die Implementierung des Ressourcen-Managers
- Begrenzt Ressourcen signifikant durch strenge Kriterien	- Manuelle Einstellungen nur durch eine Anpassung der Abfrage selbst

Table 4: Vergleich Statische und Dynamische Methode

4. Implementierung

In diesem Kapitel werden Bibliotheken, Anwendungen und Integrationsprogramme beschrieben, die hinter den Ressourcen-Manager stehen und für die Implementierung benötigt werden. Wie diese zusammenhängen, wird in Abbildung 9 dargestellt. Im ersten Unterkapitel wird die WEKA Bibliothek beschrieben. Weka wird für die Klassifikation der Daten und die Erstellung der Modelle benötigt. Als nächstes wird die JDOM Bibliothek vorgestellt. JDOM wird verwendet um XML Daten manipulieren zu können. Drittens werden Webservice erklärt und letztens wird die Implementierung des Ressourcen-Manager in IBM Websphere-Process-Server und IBM Websphere Integration Developer beschrieben.

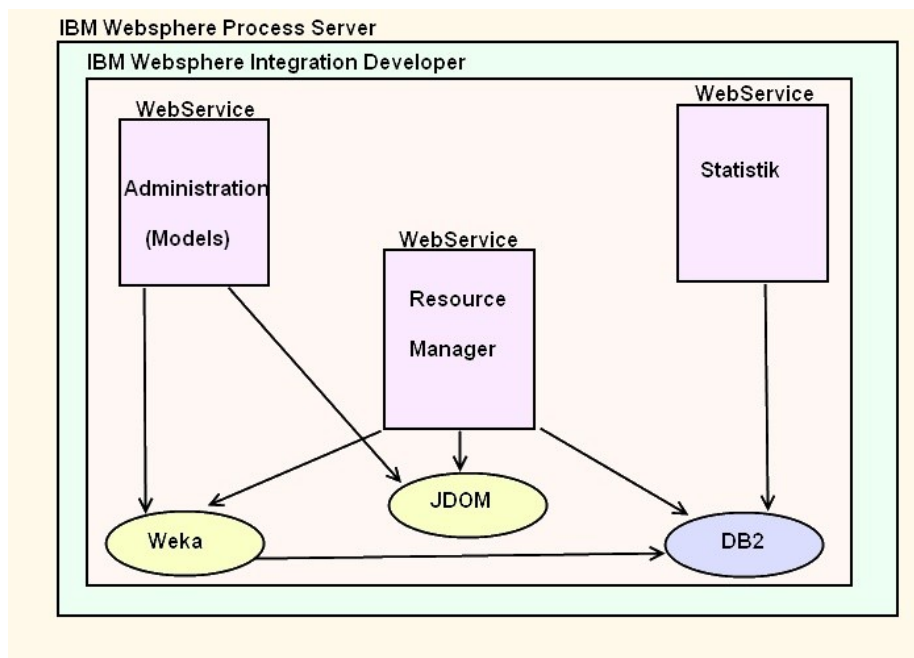


Abbildung 9: Implementierungszusammenhänge; Gelber Oval = Bibliotheken; Blauer Oval = Datenbank; Violetter Rechteck = Komponenten als Webservice; Pfeil = Komponente greift auf Bibliothek/Datenbank.

4.1 WEKA

Die Klassifikationsmodelle, welche die beste Ressource für eine bestimmte Aktivität voraussagen werden mit Hilfe der Weka [12] Bibliotheken aufgebaut. Weka wurde von der Universität Waikato aus Neuseeland entwickelt, ist Open Source, Plattform-unabhängig, in Java implementiert und ist eine Sammlung von Data Mining Algorithmen und Algorithmen für maschinelles Lernen. WEKA steht für Waikato Environment for Knowledge Analysis.

Weka stellt eine Vielzahl von effizienten Implementierungen von Data-Mining Algorithmen bereit. Die meisten sind die Klassifikationsalgorithmen. Die Klassifikatoren werden in sieben Gruppen unterteilt. Bayes, Functions, Lazy, Meta, Misc, Rules und Trees. Einige bekannte Algorithmen aus diesen sieben Gruppen sind BayesNetze, NaiveBayes, LineareRegression, MultilayerPerceptron, Kstar, M5Rules, ZeroR, M5P und viele mehr. In der Studienarbeit wird der M5P Algorithmus aus der Gruppe Trees verwendet. Neben den Klassifikatoren werden Assoziationsregel Algorithmen und Clustering Algorithmen bereitgestellt. Ein bekannter Assoziationsregel Algorithmus, der in Weka implementiert wurde und als Java Klasse zur Verfügung steht, ist der Apriori Algorithmus. Clustering Algorithmen in Weka sind SimpleK-Means, EM oder CobWeb [25].

Zur Aufbereitung der Daten wird das Filter Paket zur Verfügung gestellt. Dieses Paket ist besonders hilfreich für maschinelles Lernen. Es enthält Klassen zur Manipulation von Daten wie z.B das Hinzufügen oder Entfernen von Attributen, Klassen zur Änderung von numerischen Daten z.B durch eine benutzerdefinierte Funktion sowie die Möglichkeit, fehlende Werte zu ersetzen und viele andere [25].

Weka besitzt 4 Benutzeroberflächen. Explorer, Experimentier, Knowledge-Flow und Simple CLI sowie eine Java-API.

Die wichtigste Benutzeroberfläche ist der Explorer. Er besitzt mehrere Frontpanels zur Aufbereitung der Daten, Klassifikation, Assoziation, Clustering, Auswahl von Attributen und Visualisierung der Ergebnisse.

Der Experimentier wird eher seltener benutzt als der Explorer. Er dient dazu die Performance verschiedener Algorithmen auf Basis von verschiedenen Evaluationskriterien, welche in Weka zur Verfügung stehen, zu analysieren. Experimente sind z.B das Testen von mehreren Algorithmen auf verschiedenen Datenmengen.

Der Knowledge Flow stellt eine Alternative zum Experimentier da und besitzt einige Erweiterungen. Eine wichtige Erweiterung ist die inkrementelle Bearbeitung der Daten und ist vor allem bei großen Datenmengen hilfreich die nicht auf einmal in den Hauptspeicher passen.

Die letzte Benutzeroberfläche ist die Simple CLI. Es stellt eine Weka Shell zur Verfügung, wo Kommandos eingegeben und Antworten (Ergebnisse) visualisiert werden können. Sie ist den anderen Oberflächen überlegen, da sie vollen Zugriff auf die Weka Klassen wie Klassifikatoren, Filter, Cluster, usw. gibt [26, 27].

Die Schnittstelle zwischen den RM und Weka erfolgt über der Java-API, welche Weka zur Verfügung stellt. Die Trainings- und Testdaten werden zur Analyse oder Modelaufbau aus der Ressourcen-Manager Datenbank über JDBC geholt und weiter an WEKA zur Analyse weitergeleitet. Zur Klassifikation der Daten wird der M5P Algorithmus benutzt.

4.2 JDOM

JDOM [24] ist eine Open-Source Java Bibliothek, welche XML Daten manipulieren kann. Es ist ähnlich wie DOM (Document Object Model), ist aber nicht auf DOM aufgebaut. In der Bibliothek sind bekannte Schnittstellen integriert wie SAX (Simple API for XML) und DOM

Der RM benutzt JDOM, um auf die Workitem Daten, welche in XML Struktur als Parameter der Request und Release Funktion geschickt werden, zuzugreifen.

4.3 Webservices

Webservices [16] wurden entwickelt um die Kommunikation in ein Verteiltes System zwischen Softwarekomponenten zu erleichtern. Webservices besitzen eine Client/Server Architektur. Auf der Server Seite wird eines oder mehrere WSDL Dokumente erstellt, welche die Schnittstellinformationen wie Eingabeparameter und Ausgabeparameter enthalten. Wurde so eine WSDL Schnittstelle definiert, können mehrere Clients darauf zugreifen. Die Kommunikation zwischen den Softwarekomponente erfolgt dann via SOAP (Simple Object Acces Protokol).

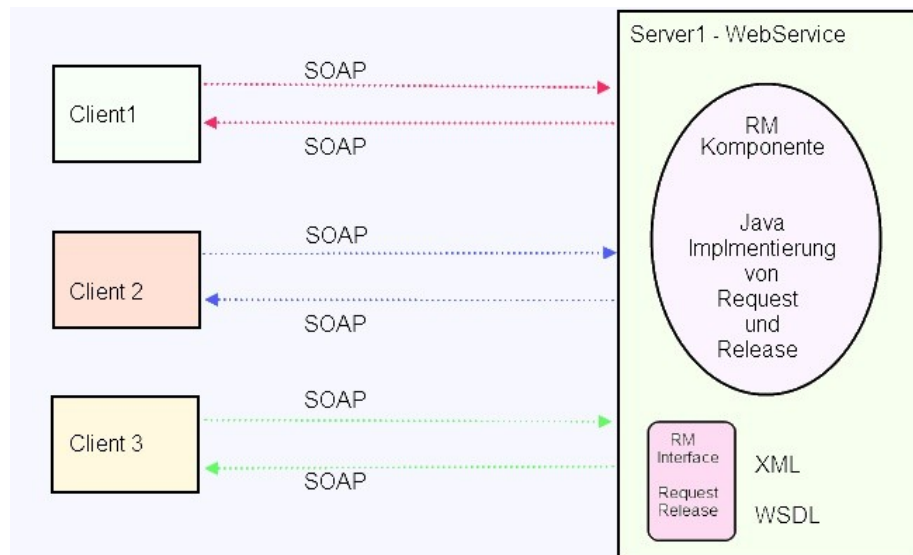


Abbildung 10: Webservice Beispiel

Der Ressourcen-Manager besitzt 3 Webservices, jedes mit eigener Schnittstelle bzw. WSDL Dokument. Dies ist in Abbildung 9 dargestellt. Diese sind :

- Der Ressourcen-Manager (RMInterface)
- Die Administrationskomponente (ModelsInterface)
- Statistikkomponente (StatistikInterface)

Der Aufbau des Ressourcen-Manager WSDL Schnittstelle wird in Abbildung 11 dargestellt. Diese Schnittstelle besitzt 2 Methoden, Request und Release, mit mehreren Eingabeparametern und einem Ausgabeparameter. Um die Daten leichter in der Datenbank speichern zu können, werden fast alle Parameter als String übergeben. Intern werden aber einige Parameter wie ResourceID oder Selection Mode in Integer gecastet. Zwei Parameter besitzen den Datentyp OutputRequest, welcher 4 Werte,

RessourceID, RequestTime, WaitingTime und Setup-Time, alle in eins zusammenfasst. Die Datentyp Definitionen werden in eine XML Schema Datei .xsd gespeichert und die Werte von Typ OutputRequest werden in XML Struktur gespeichert und geschickt.

request		
	Role	string
	type	string
	ressourceID	string
	activityName	string
Eingaben	processName	string
	processID	string
	selectionMode	string
	goalFunction	string
	workItem	string
	firstActivityName	string
Ausgaben	requestOutput	OutputRequest
release		
	fromRequest	OutputRequest
	release	string
	activityName	string
Eingaben	processName	string
	processID	string
	goalFunction	string
	tempworkItem	string
	workItem	string
Ausgaben	releaseStatus	string

Abbildung 11: Ressourcen-Manager Schnittstelle in WID

Wie eine Anfrage auf ein Webservice erfolgt, wird in Abbildung 10 dargestellt. Client X schickt eine Anfrage an den Server. Auf der Server Seite befindet sich ein WSDL-XML Dokument, wo die Schnittstelle mit den Eingabeparameter und Ausgabeparameter beschrieben ist. Die Inputdaten werden der JavaKomponente geschickt (z.B die RessourcenManager Komponente) und verarbeitet. Die Ergebnisse der Methode werden auch als WSDL-XML geschrieben und zu den Client geschickt. Die XML Kommunikation zwischen Server und Client erfolgt über SOAP und HTTP.

4.4 IBM Websphere Process Server und IBM Integration Developer

Eine Lösung, Geschäftsprozesse zu erstellen, verwalten und zu automatisieren, kann mit Hilfe der von IBM entwickelten Websphere Process Server und Websphere Integration Developer [17] geschehen. Diese zwei Produkte stellen eine Plattform zur Verfügung die konsistent und sicher ist sowie die ACID [30] Eigenschaften erfüllt. IBM Websphere Process Server basiert auf IBM Websphere Application Server und stellt eine SOA Integrations-Plattform zur Verfügung. Mit Hilfe von IBM Websphere Integration Developer können Applikationen erstellt werden welche auf IBM Websphere Process Server ausgeführt werden. Entwickler die mit Eclipse Erfahrungen gesammelt haben, werden es in IBM Integration Developer leicht haben. Der WID basiert auf Eclipse und unterstützt SOA Business Process Management Lösungen und Integrationslösungen für WPS. Viele Komponenten wie Services, Geschäftsregeln, Mediationen oder Prozesse können benutzt werden um komplexe Geschäftslösungen zusammenzustellen. Komponenten können leicht über Drag and Drop in ein Assembly-Diagramm gezogen werden und Geschäftsprozesse können visuell leicht erstellt werden.

In der Studienarbeit wird der IBM Integration Developer 7 und IBM Websphere Process Server 7 benutzt, um den Ressourcen-Manager als Webservice bereit zustellen. Abbildung 12 zeigt die Implementierung des Ressourcen-Managers und dessen Teilkomponente in IBM Websphere Integration Developer.

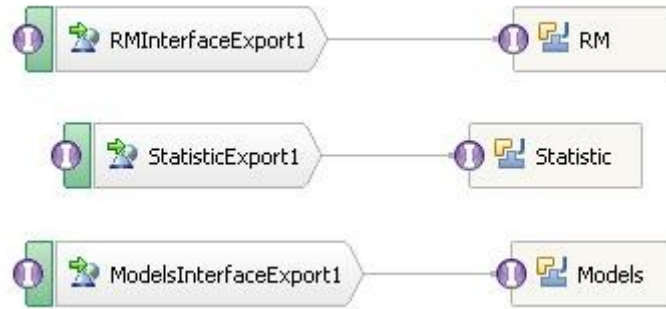


Abbildung 12: Assemblydiagramm des Ressourcen-Managers und seiner Teilkomponenten in Integration Developer

5. Der Ressourcen-Manager und die Teilkomponenten

Der Ressourcen-Manager und die Teilkomponenten wurden in Java implementiert. Die Implementierung verwendet die WEKA und die JDOM Bibliothek und besteht aus 3 wichtigen Komponenten und einer Datenbank. Der Ressourcen-Manager, die RM Datenbank, die Administrations Komponente für die Modelle und die Statistik Komponente. Der Zugriff auf die RM Datenbank erfolgt via JDBC. Die Komponenten sind als Webservice verfügbar und wurden mit Hilfe von IBM Websphere Integration Developer und IBM Websphere Process Server integriert.

In den nächsten Unterkapitel werden alle 3 Komponenten und die Datenbank erklärt. In den ersten Unterkapitel wird die Ressource Manager Datenbank diskutiert und die Beziehungen zwischen den Tabellen erklärt. Als nächstes wird die Administrationskomponente beschrieben. Hier werden temporäre Tabellen, die für die Klassifikation der Daten benötigt werden, und die M5P Bäume erstellt. Als nächstes wird die Ressourcen-Manager Komponente erklärt. Hier werden alle Schritte vom Aufruf bis zur Freigabe detailliert beschrieben. Letztens wird die die Statistik Komponente, die später für Optimierungszwecke verwendet werden kann, beschrieben.

5.1 Resource Manager Datenbank

Die Resource Manager Datenbank RM (Abbildung 9, DB2) besteht aus 14 Tabellen und wurde in das relationale Datenbankmanagementsystem DB2 implementiert. Die Beziehungen zwischen den Tabellen sind in der Abbildung 13 sichtbar. Wie man in der Abbildung erkennen kann enthält die Datenbank, Ressourcen-, Ausführungs-, Geschäftsaktivitäts- sowie Geschäftsprozessdaten. Die Tabellen können in zwei Kategorien eingeteilt werden. Ressourcen- und Ausführungsdaten. Eine Ressource besitzt immer eine Rolle. Wie die Rollen aufgebaut sind, wurde in Kapitel 1.2 erklärt. Es wurden zwei Arten von Ressourcen implementiert. Mitarbeiter (Staff) und Maschine (Machine). Die Staff Tabelle enthält persönliche Daten wie Name, Alter oder Mitarbeiterbewertungen. Die Staff Tabelle steht in Beziehung mit Field, FieldName, Degree, Training und TrainingName. Ein Mitarbeiter kann in mehrere Fachgebieten (Field) spezialisiert sein, wo er einen bestimmten Abschluss (Degree) erworben hat. Außerdem kann er mehrere Trainings abgeschlossen haben. Die Machine Tabelle enthält Daten wie Alter, Typ, Hersteller, Preis sowie Setup-Zeiten die bei der Installation einer Maschine entstehen. Die Machine Tabelle steht in Beziehung mit den Features und FeatureName Tabellen. Eine Machine kann mehrere Merkmale besitzen. Die Execution Tabelle wird für den Aufbau der Modelle und von der Statistik Komponente, erläutert in Kapitel 5.4, verwendet. Es enthält Daten über die Ausführung einer Aktivität mit einer bestimmten Ressource. Gespeichert werden Ausführungszeiten, Ausführungswartezeiten sowie bei Maschinen Ausführungssetupzeiten. Die Execution Tabelle steht mit den Process und Activity Tabellen in Verbindung. Die komplette Datenbank mit allen Attributen ist in Anhang 1 aufgelistet.

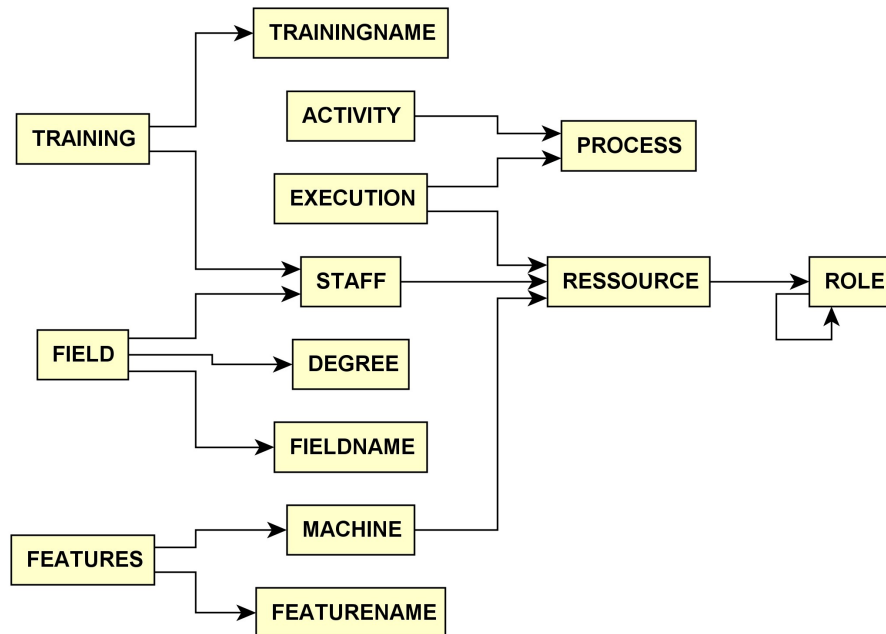


Abbildung 13: RM Datenbank

5.2 Administrationskomponente

Hier werden die Linearen Modelle aufgebaut und in der RM Datenbank in der Model Tabelle gespeichert. Der Aufruf zum Modelaufbau geschieht über die Methode `buildmodels()`, welche eine Reaktion von mehreren Funktions- und Prozeduraufrufe startet.

Als erstes werden aus der RM Datenbank, temporäre Tabellen für Ressourcen von Typ `Staff` und `Machine` erzeugt, welche nur die Daten, die wichtig für den Modelaufbau sind, verwenden. Zweitens werden die TempTabellen mit den Daten aus der RM Datenbank gefüllt. Letztens werden die Dataanalysis Tabellen erzeugt, welche direkt an die Weka Bibliothek geschickt werden, damit die M5P Bäume erzeugt werden können. Die Attribute von Typ `Staff` und `Machine` unterscheiden sich. Die Attribute werden in den nächsten Paragraphen erklärt. Abbildung 14 zeigt wie die Administrationskonsole die Modelle aufbaut.

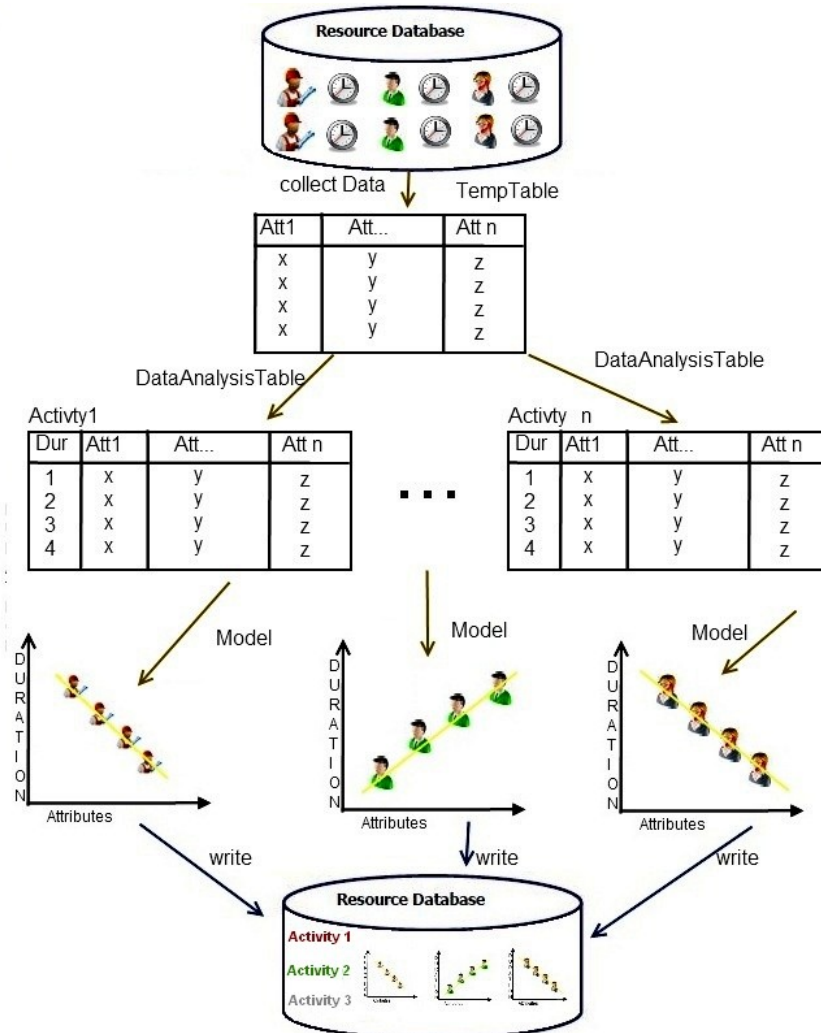


Abbildung 14: Schritte zum Aufbau der Modelle

Die StaffTempTable Tabelle enthält die meisten Attribute der Staff Tabelle und der Tabellen die mit der Staff Tabelle über einen Foreign Key in Verbindungen stehen.

Die MachineTempTable Tabelle enthält die meisten Attribute der Machine Tabelle und der Tabellen die mit der Machine Tabelle über ein Foreign Key in Verbindungen stehen.

Als nächstes werden die oben beschriebenen Tabellen mit den Daten aus der RM Tabelle gefüllt. Damit die Ausbildungs- und Trainingsdaten eine mathematische Bedeutung in den Modellen besitzen sollen, müssen diese Daten umgewandelt werden. Hat der Mitarbeiter eine gewisse Ausbildung oder Training oder die Maschine

typische Features werden diese an der entsprechenden Stelle mit einer 1 belegt. z.B hat der Mitarbeiter X, Mathematik studiert erscheint in der StaffTempTable in der Spalte Mathematik eine 1, sonst 0.

StaffID	FieldID
1	1
1	2
2	2

FieldID	Description
1	Mathematik
2	Biologie

Table 6: Ausbildungsbeschreibungen

Table 5: Welche Ausbildung hat welcher Mitarbeiter

Daraus entsteht

StaffID	Mathematik	Biologie
1	1	1
2	0	1

Table 7: Aus Tabelle 5 und 6 resultierende Mitarbeitertabelle

Trainings-, Ausbildungs- und Featuresdaten werden nach diesen Algorithmus befüllt.

Der nächste Schritt bereitet die Daten für die Analyse vor. Für jede Aktivität wird eine DataAnalysisTabelle erzeugt, die die Execution Werte + TempTable Werte + (optional) WorkItem Werte enthält. Wenn Workitems verlangt sind, erweitert man die Spalten um die Elemente der Workitems. Beispiel :

Workitems haben folgende Struktur.

```

<Name>
<Workitem_1> Wert_1 </Workitem_1>
...
<Workitem_n> Wert_n </Workitem_n>
</Name>

```

Mit Hilfe von JDOM werden die Kinderelemente und deren Werte von der Wurzel ausgewählt. Als erstes wird geprüft welche Datentypen die Workitems haben. Möglich Datentypen sind : String, Numeric oder Boolean. Diese werden für DB2 folgend definiert : VarChar, Integer, Smallint (0,1). True = 1; False = 0.

Die neuen Spalten erweitern sich wie in der unteren Tabelle zu sehen ist.

Vorhandene Attribute	WorkItem_1	...	Workitem_n
----------------------	------------	-----	------------

Table 8: Tabelle 7 plus Workitems

Alle Werte die in der DataAnalysis Tabellen enthalten sind, werden an WEKA zum Modelaufbau geschickt. Die Linearen Modelle werden in einer Hashtabelle gespeichert wo der Aktivitätsname der Schlüssel ist und die Linearen Modelle die Werte.

Insgesamt baut man 4 HashTabellen :

- Hash_Data_Current_Activity_Without_WorkItem
- Hash_Data_Current_Activity_With_WorkItem
- Hash_Data_All_Activities_Without_WorkItem
- Hash_Data_All_Activities_With_WorkItem

Die HashTabellen mit den Modellen werden in der RM Datenbank in der Model Tabelle gespeichert.

5.3 Ressource Manager

Der Ressourcen Manager ist die wichtigste Komponente. Er ist zuständig dafür welche Ressource für eine bestimmte Aktivität gewählt wird, sucht welches Model für die Anfrage benutzt wird und gibt Ressourcen wieder frei oder speichert sie als besetzt für den Prozess der sie benötigt. Zusätzlich speichert er auch Ausführungszeiten die für die Optimierung und den Modelaufbau wichtig sind. Der RM besitzt 2 Funktionen auf welche man von außen zugreifen kann. Funktion Request und Funktion Release. Alle Schritte des Ressourcen Managers werden in Abbildung 15 dargestellt und auf die Funktionen wird in den nächsten Unterkapiteln eingegangen.

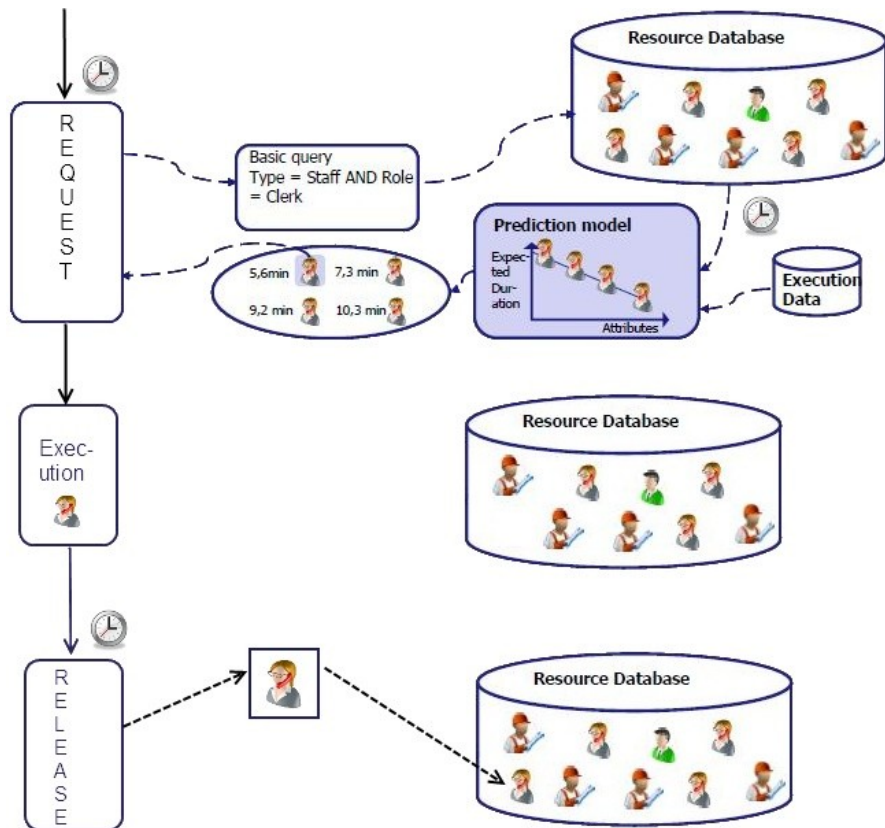


Abbildung 15: Ressourcen-Manager Verlauf

5.3.1 Request

Die Request Funktion wählt die Ressource aus der Datenbank, die die Eigenschaften der Request-Parameter erfüllen und speichert sie als besetzt. Die Allokation der Ressourcen ist ein Mehrschrittverfahren. Die Parameter und das Allokationsverfahren wird in den nächsten 2 Unterkapitel erklärt.

5.3.1.1 Parameter

Die Übergabeparameter der Request Funktion bestimmen, welche Ressource aus der Datenbank gewählt wird. In der unteren Tabelle werden die Parameter erklärt :

Name	Beschreibung
Type	<ul style="list-style-type: none"> - Beschreibt den Ressourcentyp und kann entweder den Wert „Staff“ oder „Machine“ annehmen
Role	<ul style="list-style-type: none"> - Die Role definiert welche Position aus der Role-Hierarchy für die Aktivität benötigt wird. - Alle möglichen Role Werte kann man aus der ROLE Tabelle des RM entnehmen
RessourceID	<ul style="list-style-type: none"> - Wird der Wert 0 übergeben, muss eine neue Ressource ausgewählt werden - Wird ein Wert > 0 geschickt, meint man damit eine genaue Ressource, die gewählt sein soll
ActivityName	<ul style="list-style-type: none"> - Beschreibt den Aktivitätsnamen der die Ressource in Anspruch nimmt
ProcessName	<ul style="list-style-type: none"> - Beschreibt den Prozessnamen der die Ressource in Anspruch nimmt
ProcessID	<ul style="list-style-type: none"> - Beschreibt eine eindeutige ProzessID welche die Ressource in Anspruch nimmt
SelectionMode	<p>Definiert welcher Algorithmus zur Ressourcenauswahl genommen werden sollte :</p> <ul style="list-style-type: none"> Wert 1 = Zufällig (Random) Wert 2 = M5P
GoalFunction	<p>Definiert welche Zielfunktion gewählt werden soll :</p> <ul style="list-style-type: none"> - Current_Activity : Optimiert die Ressource Selektion, indem die Ressourcen Optimierungszielwerte für die jetzige Aktivität benutzt wird - All_Activities : Optimiert die Ressource Selektion indem die Ressourcen Optimierungszielwerte für alle Aktivitäten benutzt werden
WorkItem	<ul style="list-style-type: none"> -XML CDATA welche Aktivitätsspezifisch ist.

	- Kann benutzt werden um die Ressourcen Selektion weiter zu verfeinern. z.B. Welche Ressource führt am besten Aktivität X aus mit den zusätzlichen Werten aus WorkItem.
firstActivityName	In ein Geschäftsprozessszenario kann es vorkommen das eine Ressource für mehrere Aktivitäten beansprucht wird. Damit man die Ausführungswerte für All Activities leicht aggregieren kann, schickt man auch den Wert der ersten Aktivität, welche die Ressource benutzt oder benutzt hat, mit.

Table 9: Parameter Request Funktion

5.3.1.2 Requestverlauf

Der Verlauf des gesamten Requests wird in Abbildung 16 dargestellt. Für jede Ressourcen-Anfrage wird als erstes die Anfragezeit gespeichert. Falls ResourceID=0, wählt man alle Ressourcen, die die Typ und Rolle Bedingungen erfüllen. Falls ResourceID > 0 wählt man die angegebene Ressource. Wenn wir annehmen, dass die Parameterwerte richtig geschickt wurden und dabei keine Ressourcen zurückgeliefert wurden, bedeutet dass diese nicht verfügbar sind und gewartet werden muss bis mindestens eine wieder frei ist. Die einzige Möglichkeit eine nicht verfügbare Ressource auszuwählen ist, wenn sie für einen Prozess blockiert wurde und eine Aktivität des gleichen Prozesses sie wieder verlangt. Falls $x:x > 1$ Ressourcen die Bedingungen erfüllen ist der nächste Schritt eine daraus auszuwählen. Der RM verfügt über 2 Selektionsverfahren. Ist SelectionMode = 1, wählt der RM zufällig eine Ressource aus. Ist SelectionMode = 2 wählt der RM ein Modell aus. Wenn zusätzlich noch Workitems geschickt werden, werden Attributwerte und Workitems zur Klassifizierung geschickt, sonst werden nur Attributwerte geschickt. Nachdem die Werte Klassifiziert worden sind, betrachtet man die vom den M5P Algorithmus zurückgelieferten Werte und der kleinste wird ausgewählt. Der kleinste Wert bedeutet der schnellste. Das beste Ergebnis ist in unserem Fall die Ressource welche die Aktivität am schnellsten, nach der Voraussage des Modells, ausführen würde. Insgesamt gibt es 4 mögliche Modelltypen. Jedes Modelltyp ist für jede Aktivität aufgebaut. Dabei unterscheidet man zwischen:

- Current_Activity_Without_Work_Item
- Current_Activity_With_Work_Item
- All_Activities_Without_Work_Item
- All_Activities_With_Work_Item

Welches Model genommen werden soll entscheiden die Request-Parameter. Falls WorkItem Parameter NULL ist wählt man die *_Without_Work_Item Modelle und *=Paramterwert von GoalFunction.

Wurde eine Ressource ausgewählt, wird sie als nicht verfügbar in der Datenbank gespeichert. Als nächstes schaut man ob der Type von Typ Maschine ist. Wenn ja wird eine Setup-Time ausgeführt. Als letztes gibt die Request Funktion alle Werte die erzeugt wurden zurück. Diese sind RessourceID, StartRequestTime, WaitingTime und SetupTime.

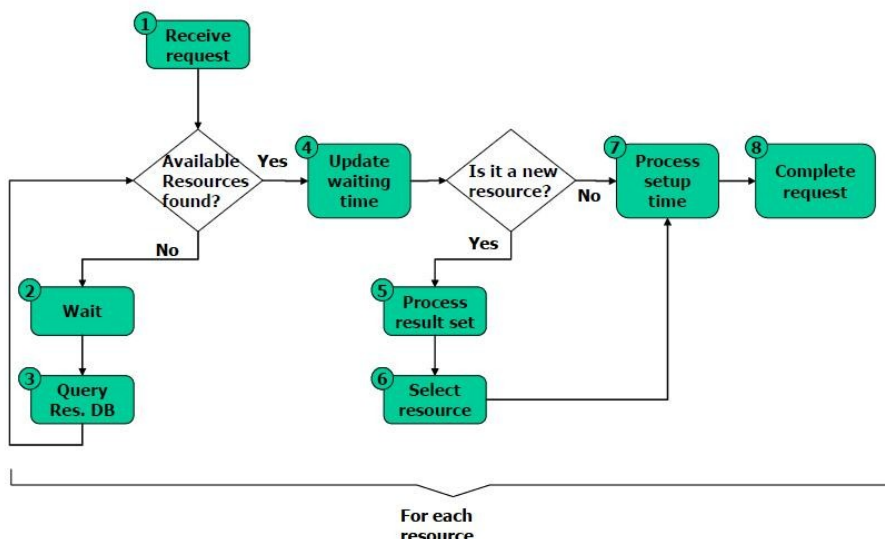


Abbildung 16: Request verlauf [1]

5.3.2 Release

Der Release ist dafür zuständig, Ressourcen komplett freizugeben oder sie nur teilweise freizugeben. Teilweise Freigabe bedeutet, dass Aktivitäten desselben Prozesses noch die Möglichkeit haben, die teilweise freigegebenen Ressource in Anspruch zu nehmen. Die Parameter und das Allokationsverfahren wird in den nächsten zwei Unterkapitel erklärt.

5.3.2.1 Parameter

Name	Beschreibung
FromRequest	<p>- Enthält die Werte, die im Request berechnet wurden</p> <ul style="list-style-type: none"> • RessourceID • StartRequestTime • WaitingTime • SetupTime
Release	<p>- Entscheidet ob eine Ressource komplett oder teilweise Freigegeben wird</p> <p>1 = Komplette Freigegebenen</p> <p>2 = Freigegeben nur für Aktivitäten derselben ProzessInstance</p>
ActivityName	- Beschreibt den Aktivitätsnamen der die Ressource in Anspruch nimmt
ProcessName	- Beschreibt den Prozessnamen der die Ressource in Anspruch nimmt
ProcessID	- Beschreibt eine eindeutige ProzessID welche die Ressource in Anspruch nimmt
GoalFunction	<p>Definiert welche Zielfunktion gewählt werden soll :</p> <p>2 Mögliche Werte :</p> <ul style="list-style-type: none"> - Current_Activity : Optimiert die Ressource Selektion indem die Ressourcen Optimierungszielwerte für die jetzige Aktivität benutzt werden - All_Activities : Optimiert die Ressource Selektion indem die Ressourcen Optimierungszielwerte für alle Aktivität benutzt werden
firstActivityName	In ein Geschäftsprozessszenario kann sein das eine Ressource für mehrere Aktivitäten beansprucht wird. Damit man die Ausführungswerte für All_Activities leicht aggregieren kann, schickt man auch den Wert der ersten Aktivität welche die Ressource benutzt oder benutzt

	hat mit.
WorkItem	<ul style="list-style-type: none"> -XML CDATA welche Aktivitätsspezifisch ist. - Kann benutzt werden um die Ressourcen Selektion weiter zu verfeinern . z.B Welche Ressource führt am besten Aktivität X aus mit den zusätzlichen Werten aus WorkItem

Table 10: Parameter Release Funktion

5.3.2.2 ReleaseVerlauf

Der Verlauf des Releases beinhaltet viel weniger Schritte als der Request und wird in Abbildung 17 dargestellt. Zuerst wird überprüft ob ReleaseMode=1 (vollständiges Release) oder ReleaseMode=2 (teilweise Freigabe).

Wenn ReleaseMode=1, werden in der Ressource Tabelle zwei Veränderungen durchgeführt. Als erstes wird die entsprechende Ressource wieder als verfügbar gespeichert (Available=1). Danach wird die ProzessID auf 0 gesetzt was bedeutet das auch eine andere ProzessInstance auf die Ressource zugreifen kann.

Wenn ReleaseMode=2, werden in der Ressource Tabelle auch zwei Veränderungen durchgeführt. Als erstes wird die entsprechende Ressource wieder als Verfügbar gespeichert (Available=1). Danach wird die ProzessID auf die jetzige ProcessInstanceID gesetzt, was bedeutet das sie nur für Aktivitäten der gleichen ProcessInstanceID wieder verwendet werden kann, sonst nicht.

Als nächstes speichert man die Dauer (RequestTime) die die Ressource gebraucht hat die angegebene Aktivität auszuführen. Von Request bis Release.

Der letzte Schritt ist verantwortlich für die Speicherung der Ausführungsdaten in der Execution Tabelle der RM Datenbank.

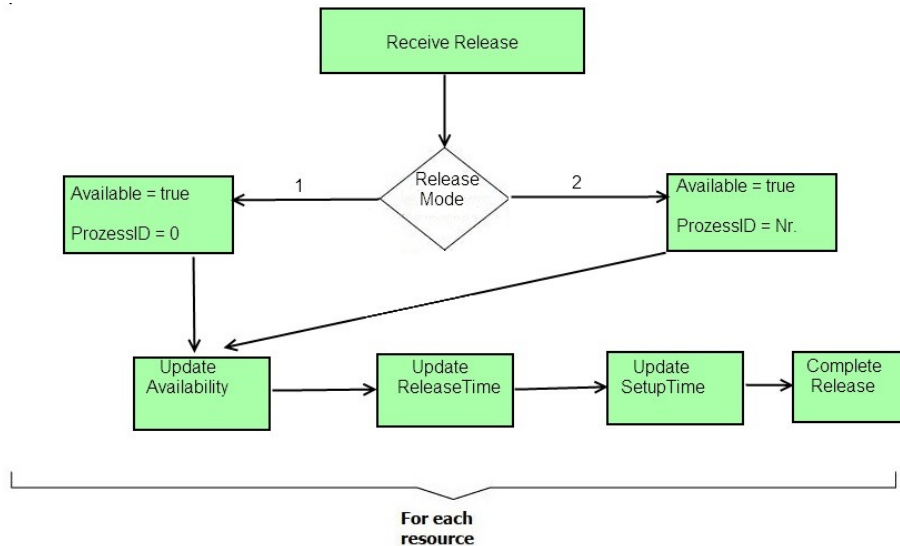


Abbildung 17: Releaseverlauf

5.4 Statistik/Metriken

Diese Komponente ist sehr wichtig für die Optimierungsschicht der dBOP Plattform. Möchte man die Geschäftsprozesse und die Modelle optimieren, muss man Informationen sammeln, wie gut die Ressourcen eine Aktivität innerhalb eines Geschäftsprozesses ausgeführt haben. Dabei untersucht man mehrere Ausführungsdaten wie z.B. Arbeitsdauer, Wartezeit oder Verfügbarkeit.

Insgesamt wurden 8 Funktionen implementiert, auf welche man über die Statistikkomponente zugreifen kann. Dies geschieht indem eine der Funktionen ausgewählt wird und die Parameter (Unterkapitel 5.4.1 und 5.4.2 Parameter Spalte) in der richtigen Reihenfolge geschickt werden.

Die nächsten zwei Unterkapitel erläutern Ressourcenspezifische und Aktivitätsspezifische Metriken. Der letzte Unterkapitel 5.4.3 beschreibt kurz wie die gewonnenen Metriken auf 5.4.1 und 5.4.2 in der Statistic Tabelle gespeichert werden.

5.4.1 Ressourcenbezogene Metriken

Ressource Metriken liefern aggregierte Informationen entweder zu einer über Parameter übergebe Ressource oder Rollen.

Metrik	Beschreibung	Varianten	Parameter
Overall Utilization	Berechnet, wie lange eine Ressource in ein gewissen Zeitraum beansprucht wurde	SUM, AVG, MIN, MAX	ResRole (ResourceID oder Rolle), StartTime, EndTime, AnalysisOption (Varianten)
Productive Utilization	Berechnet, wie lange eine Ressource gearbeitet hat	SUM, AVG, MIN, MAX	ResRole (ResourceID oder Rolle), StartTime, EndTime, AnalysisOption (Varianten)
Waiting Time	Gibt an, wie lange auf eine Ressource gewartet werden musste	SUM, AVG, MIN, MAX	ResRole (ResourceID oder Rolle), StartTime, EndTime, AnalysisOption (Varianten)
Bottle Neck	Berechnet, wie oft eine Ressource eine Aktivität nicht sofort ausführen konnte	COUNT	ResRole (ResourceID oder Rolle), StartTime, EndTime
Availability	Indiziert wie oft eine Ressource sofort nach Auswahl die Ausführung eine Aktivität starten konnte	COUNT	ResRole (ResourceID oder Rolle), StartTime, EndTime

Table 11: Parameter / Beschreibung Ressourcenbezogene Metriken

5.4.2 Aktivitätsbezogene Metriken

Aktivität Metriken liefern Informationen entweder zu einer über Parameter übergebe Aktivität.

Metrik	Beschreibung	Varianten	Parameter
Waiting Time	Indiziert wie lange auf eine Aktivität auf eine Ressource warten musste	SUM, AVG, MIN, MAX	ResRole (ResourceID oder Rolle), StartTime, EndTime, AnalysisOption (Varianten)
Bottle Neck	Berechnet wie oft eine Aktivität auf eine Ressource warten musste und nicht sofort ausgeführt werden konnte	COUNT	ResRole (ResourceID oder Rolle), StartTime, EndTime
Availability	Indiziert wie oft eine Aktivität sofort nach Auswahl, die Ausführung mit einer Ressource starten konnte	COUNT	ResRole (ResourceID oder Rolle), StartTime, EndTime

Table 12: Parameter / Beschreibung Aktivitätsbezogene Metriken

5.4.3 Statistic Tabelle

Die Statistik Tabelle sammelt alle Informationen die aus 5.4.1 und 5.4.2 gewonnenen wurden. Benutzer können somit leicht auf die Ergebnisse der Analyse zugreifen und sie auf verschiedenen Ebenen weiter aggregieren.

Die Statistik Tabelle enthält die Spalten Analysisistime, Analysisbasedtype, Analysisfunction, Analysisstyp, Analysisressource, AnalysisValue und Analysismethod.

Wobei Analysisistime die Zeit des Aufrufes speichert. Analysisbasedtype enthält entweder den Wert Ressource oder Aktivität. Analysisfunction beschreibt den Namen der oben definierten Funktionen. Analysisstyp speichert folgende mögliche Werte, Role, Ressource, activityName. Analysisressource speichert die Namen wie Mechanic z.B für Role, oder den Wert 8 für Ressource. AnalysisValue enthält das Ergebnis das von der Funktion zurückgeliefert wurde. Analysismethod speichert folgende mögliche Werte : SUM,AVG,MIN,MAX,COUNT.

6. Evaluation

Anhand eines vereinfachten Geschäftsprozess-Beispiels werden die Vorteile des Ressourcen Managers mit Hilfe des modellbasierten Ansatzes gezeigt. Die Testergebnisse werden mit verschiedenen Modelleinstellungen untersucht. Zuerst wird der Beispielentwurf eines Geschäftsprozesses und der Aufbau erklärt. Danach werden die Ergebnisse, die gewonnen wurden, erläutert.

6.1 Evaluation Entwurf

Der zu untersuchende Geschäftsprozess, welcher in Abbildung 18 dargestellt wird, ist aus dem Versandhandel.

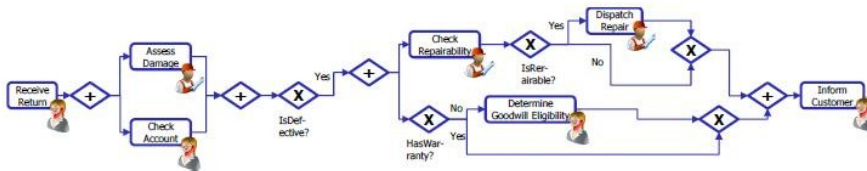


Abbildung 18: Geschäftsprozessbeispiel [1]

Wie schnell eine Ressource eine Aktivität aus dem Geschäftsprozess ausführt, hängt stark von dessen Attribute wie Ausbildung oder Trainings ab. Weil der obige Entwurf mehrere Geschäftsaktivitäten, die unterschiedliche Ressourcenrollen enthalten und entlang des Prozesses eine Ressource mehrmals in Anspruch genommen werden kann, ist es eine gute Testumgebung für den Ressourcen Manager. Bei der Evaluierung werden zwei Aspekte untersucht. Erstens wie die Performance eines Prozesses sich mit unterschiedlichen Optimierungszielen verhält. Die Tests wurden mit zwei verschiedenen Selektionseinstellungen (Random und MSP) und den zwei Zielfunktion (Current_Activity, All_Activities) durchgeführt. Zweitens wird die Annahme untersucht, dass der modellbasierte Ansatz bei steigender Anzahl von verfügbaren Ressourcen bessere Resultate erzielt. Zu diesen Zweck, wurde der Entwurf folgend aufgebaut :

- **Szenarien** : Neun verschiedene Szenarien wurden gemessen, indem zwei Dimensionen kombiniert wurden. Die erste Dimension ist die Optimierungsebene mit folgende Auswahlmöglichkeiten :
 - ◆ keine Optimierung
 - ◆ MSP mit Current_Activity
 - ◆ MSP mit All_Activities

Die zweite Dimension ist die Anzahl der verfügbaren Ressourcen während der Evaluierung.

- ◆ Niedrig (> 80% benutzt)

- ◆ Mittel
- ◆ Hoch (< 20% benutzt)
- **Data** : Insgesamt wurden 3000 Prozess Threads gestartet. Die Modelle wurden mit 1200 von diesen Daten aufgebaut, ohne irgendwelche Optimierungseigenschaften aktiviert zu haben. Die Evaluation hat mit 600 Threads pro Optimierungszielgruppe stattgefunden. Es gab 3 Optimierungszielgruppen, Random, Current_Activity und All_Activities. Insgesamt also 1800 Threads.
- **Erfolgsmessung** : Wie erfolgreich die Ressourcen mit verschiedenen Optimierungseigenschaften waren, kann man Anhand der Zeitdauer messen [1].

6.2 Ergebnisse

Die durchgeführten Tests lieferten die Ergebnisse, welche in Abbildung 19 zu sehen sind. Wie man deutlich in der Abbildung sehen kann, bringt der Modellbasierte Ansatz (Current_Activity, All_Activities) im Durchschnitt bessere Ergebnisse als der zufällige Ansatz, wo keine Optimierung vorhanden ist. Zusätzlich zeigen die All_Activities Modelle bessere Ergebnisse als die Current_Activity Modelle.

Die Annahme das der modellbasierte Ansatz bei steigender Anzahl von verfügbaren Ressourcen bessere Resultate erzielt, ist auch durch die Ergebnisse bestätigt worden. Wie man bei den modellbasierten Ansätzen (Current_Activity, All_Activities) zu sehen ist, zeigen die durchgeführten Tests, dass Ressourcen mit hoher Verfügbarkeit, bessere Ergebnisse (14%-22% besser) als Ressourcen die niedriger Verfügbarkeit erzielen [1].

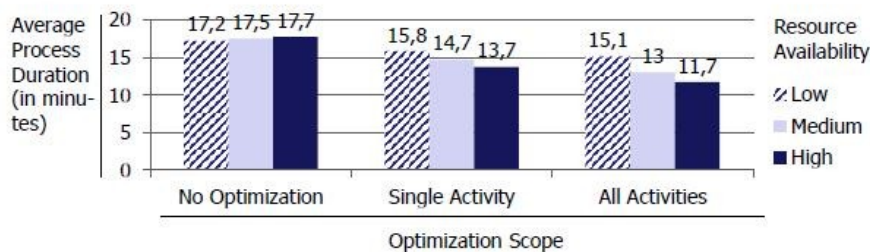


Abbildung 19: Evaluationsergebnisse [1]

7. Abschluss und Ausblick

Der letzte Kapitel beschreibt in Unterkapitel 7.1 mögliche Erweiterungen, welche den Ressourcen-Manager verbessern könnte und ihm noch stärker Realitätsnah machen zu lassen. Letztens wird die Studienarbeit zusammengefasst.

7.1 Mögliche Erweiterungen

Für den Ressourcenmanager gibt es derzeit vier Erweiterungsmöglichkeiten.- Die erste ist die automatische Entlastung einer sehr oft beanspruchten Ressource. Zweitens kann man die Optimierungsziel Funktion auf mehrere Dimensionen erweitern. Drittens die Automatisierung des Modelaufbaus und letztens die Einführung von Setup-Zeiten bei Änderung der Workitems. Wie man das erreicht und wie der Ressourcen-Manager auch auf andere Ebenen verbessert werden kann wird in den nächsten Paragraphen erläutert :

- Um eine Aktivität schnell auszuführen wählt die Implementierung immer die beste Ressource, die verfügbar ist, aus der RM Datenbank aus. Angenommen die Aktivität B_2_X (Aktivität p_i_r = Aktivität i von Prozessinstanz p wählt Ressource r) startet wenn Aktivität A_1_X beendet ist, bedeutet das Aktivität B_2_X wieder die Ressource X in Anspruch nimmt. In einem solchen Zyklus ist die Ressource X sehr stark beansprucht, während die anderen Ressourcen vielleicht ruhen. Wenn die Ressource X , k -mal beansprucht wurde, könnte man automatisch die Ressource X als nicht Verfügbar deklarieren und sie erst nach n -Schritten wieder freigeben.

- Die Optimierungsziel Funktion ist beschränkt auf die Dauer. Neben der Dauer können aber auch andere Optimierungsziel Funktion eingesetzt werden, wie Kosten oder Flexibilität. Abbildung 20 zeigt, wie die Modelle auf mehrere Optimierungsdimensionen erweitert werden könnten. Immer die beste Ressource auszuwählen, ist auf Dauer kostspielig. Eine Aktivität, die von vielen Mitarbeitern durchgeführt werden kann, ohne spezielle Trainings- oder Ausbildungskurse besucht zu haben wie Dokument drucken oder Tastatur anschließen, muss nicht unbedingt von den besten Mitarbeiter durchgeführt werden, sondern kann von den Mitarbeiter mit den geringsten Kosten erledigt werden. Beispiel : Man ruft nicht den Mitarbeiter aus dem Stock 4, um eine Tastatur an ein Computer in Stock 1 anzuschließen, wenn es in Stock 1 Personen gibt die diese Tätigkeit durchführen könnten. Die Flexibilität beschreibt wie schnell ein Mitarbeiter oder eine Maschine einsetzbar ist. Braucht man schnell die Fertigung eines bestimmten Produktes und die Einrichtzeit der besten Maschine ist sehr hoch, kann man eine Maschine wählen, die schnell eingerichtet ist, aber an der Schnelligkeit etwas leidet.

- Eine weitere Erweiterung wäre die Automatisierung des Modelaufbaus. Es stehen zwei Möglichkeiten zur Verfügung. Erstens besteht die Möglichkeit, die Modelle jedes mal, wenn ein Request-Aufruf startet, neu zu erstellen. Die Zeitkomplexität würde aber darunter leiden. Der zweite Ansatz wäre die Modelle nach bestimmten Kriterien aufzubauen. Zum Beispiel nach einer gewissen Zeit oder wenn eine bestimmte Anzahl an Aktivitäten ausgeführt worden sind.

- Die Workitems in ein Produktionsprozess können sich ändern und die Vorgänge können stark variieren. Änderungen können manchmal nicht so leicht vorge-

nommen werden und benötigen Setup Zeiten. Erweiterungen sollten eine Kostenfunktion einbauen, welche das Bewegen von Workitem zu einer bestimmten Ressource berechnet, eine Wertfunktion für Workitem sowie auch Setup Zeiten mitberücksichtigt [1].

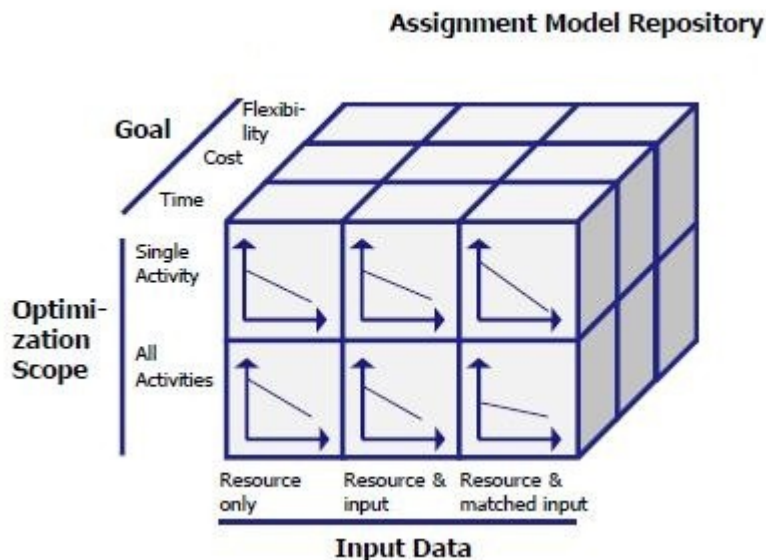


Abbildung 20: Dimensionen [1]

7.2 Zusammenfassung

In dieser Studienarbeit wurde gezeigt, wie Geschäftsprozesse mit Hilfe der richtigen Auswahl von Ressourcen optimiert werden können. Richtige Auswahl bedeutet diejenige Ressource auszuwählen die mit einer hohen Wahrscheinlichkeit die Geschäftsaktivität am schnellsten ausführen würde. Dadurch kann viel Zeit gespart werden und die gesparte Zeit kann benutzt werden, um andere Aktivitäten auszuführen.

Unternehmen können somit sich Wettbewerbsvorteile gegenüber der Konkurrenz schaffen und sich ein schnelleres Wachstum des Umsatzes und der Kunden- bzw. Lieferantenzufriedenheit erhoffen.

In der Evaluation, anhand eine typischen Geschäftsprozessbeispiels, wurde gezeigt, dass der Modell-basierte Ansatz, Ressourcen auszuwählen, dem normalen, zufälligen Ansatzes überlegen ist. Jedes Attribut kann eine entscheidende Rolle haben, wie schnell eine Ressource eine Geschäftsaktivität ausführt. Typische Ressourcen Attribute wie Ausbildung, Training oder Maschinenfeatures entscheiden somit über die Qualität, Effektivität und Schnelligkeit einer Aktivitätsausführung.

Um bestmögliche Ergebnisse zu erzielen, muss ein robustes und qualitatives Klassifikationsmodell gewählt werden. Ein Algorithmus, der das bietet, ist der M5P Algorithmus. Ein Vorteil ist, das M5P gut mit nichtlinearen Daten umgehen kann, in-

dem er die Daten in kleinere Datenmengen partitioniert, daraus ein Baum erstellt und die Linearen Regressionsmodelle in den Blättern speichert.

Der Ressourcen-Manager kann in einigen Punkten wie die Einführung neuer Optimierungszielfunktionen wie Kosten oder Flexibilität oder die Entlastung einer sehr oft beanspruchten Ressource erweitert werden. Weitere Erweiterungen können die Berücksichtigung von Setup-Zeiten bei Änderung von WorkItems sein, sowie die Automatische Verwaltung der Modelle sein.

Auf der Basis dieser Studienarbeit wurde die Publikation [1] erstellt, welche im wesentlichen eine kompakte Form der hier vorgestellten Inhalte repräsentiert.

Literaturverzeichnis

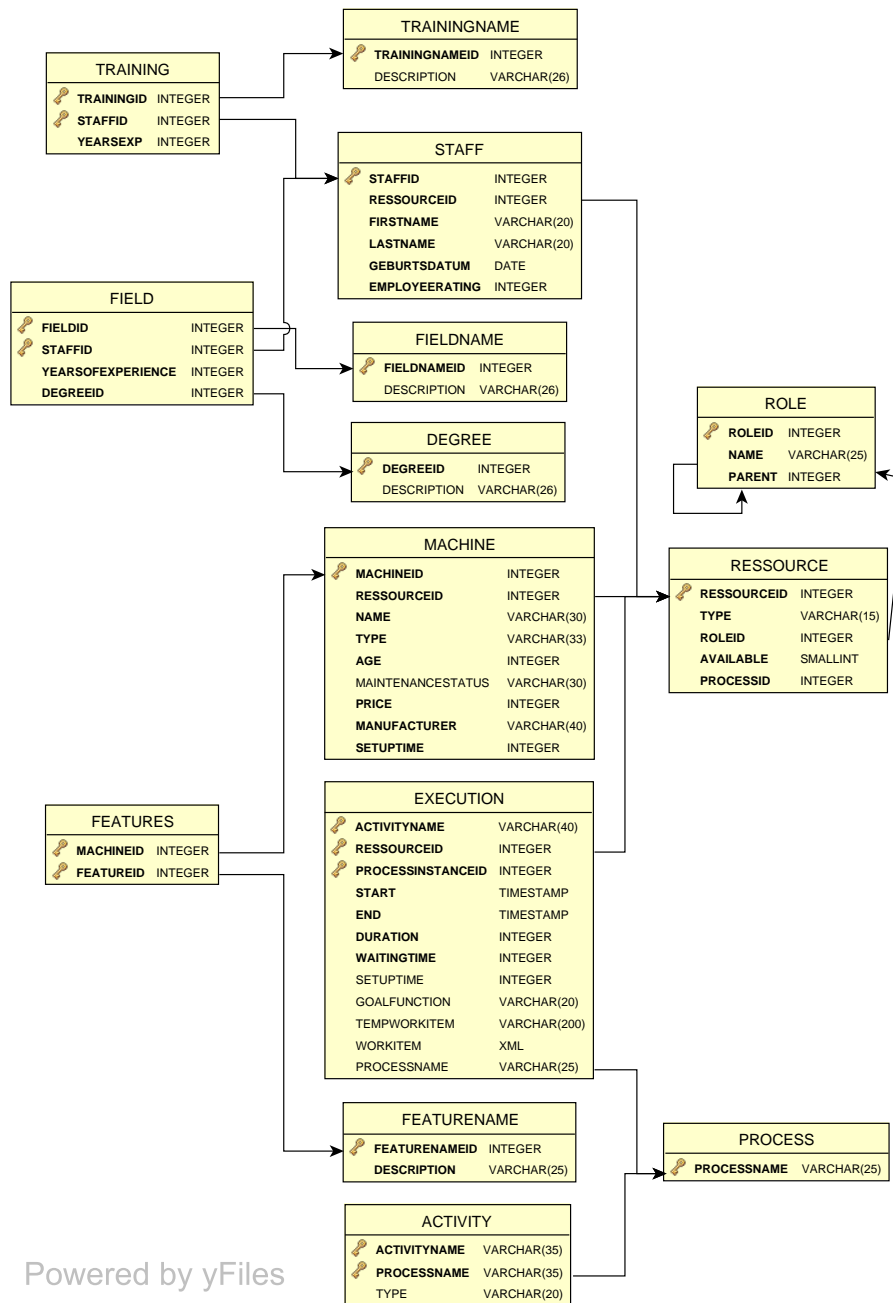
- [1] Florian Niedermann, Alexandru Pavel, Bernhard Mitschang Beyond Roles : Prediction Model-based Process Resource Management, 4th International Conference on business process and services computing (BPSC 2011)
- [2] F. Niedermann, B. Maier, S. Radeschütz, H. Schwarz, and B. Mitschang. Automated Process Decision Making based on Integrated Source Data. In Proceedings BIS 2011, 2011.
- [3] F. Niedermann, S. Radeschütz, and B. Mitschang. Deep Business Optimization: A Platform for Automated Process Optimization. In Proceedings BPSC 2010, 2010.
- [4] F. Niedermann, S. Radeschütz, and B. Mitschang. Business Process Optimization using Formalized Patterns. In Proceedings BIS 2011, 2011.
- [5] J. Han and M. Kamber. Data mining: concepts and techniques. Morgan Kaufmann, 2006.
- [6] J.R. Quinlan. Learning with continuous classes. In 5th Australian joint conference on artificial intelligence, 1992.
- [7] Y. Liu, J. Wang, Y. Yang, and J. Sun. A semi-automatic approach for workflow staff assignment. Computers in Industry, 59(5):463–476, 2008.
- [8] L. Yingbo, W. Jianmin, and S. Jiaguang. A machine learning approach to semi-automating workflow staff assignment. In Proceedings of the 2007 ACM symposium on Applied computing, 2007.
- [9] J. Anvik, L. Hiew, and G.C. Murphy. Who should fix this bug? In Proceedings of the 28th international conference on Software engineering, pages 361–370. ACM, 2006.
- [10] S. Jablonski and R. Talib. Agent assignment for process management: agent performance evaluation. In Proceedings FIT2009, 2009.
- [11] L. Ly, S. Rinderle, P. Dadam, and M. Reichert. Mining staff assignment rules from event-based data. In Business Process Management Workshops, pages 177–190. Springer, 2006.
- [12] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The WEKA data mining software: An update. ACM SIGKDD Explorations Newsletter, 11(1):10–18, 2009.

- [13] Yong Wang, Ian H. Witten : Induction of Model Trees for Predicting Continuous Classes. Working Paper 96/23 October 1996
- [14] Thomas H. Wonnacott, Ronald J. Wonnacott : Regression : A Second Course in Statistics, John Wiley & Sons Inc, 1935
- [15] Michael Berthold, David J. Hand : Intelligent Data Analysis An Introduction Second Edition, Springer Verlag 2003
- [16] Judith M. Myerson : Web Service Architectures, Tect, 2002
- [17] IBM, Auf den Seiten von IBM : IBM WebSphere Process Server 6.2 IBM WebSphere Integration Developer 6.2, <ftp://ftp.software.ibm.com/software/emea/de/web-sphere/WebSphere-Process-Server-V6-2.pdf>, Zugriff am 03.06.2011
- [18] Friedrich Rosenkranz : Geschäftsprozesse : Modell- und computergestützte Planung Zweite Auflage 2006
- [19] Mathias Weske : Business Process Management: Concepts, Languages, Architectures
- [20] Business Process Management: 4th International Conference, BPM 2006, Vienna, Austria, September 5-7, 2006
- [21] J. Becker, C. Mathas, A. Winkelmann : Informatik im Fokus : Geschäftsprozesse, Springer Verlag, 2009
- [22] S. Radeschütz, F. Niedermann, and W. Bischoff. BIAEditor - Matching Process and Operational Data for a Business Impact Analysis. Proceedings EDBT, 2010.
- [23] University of Toronto, Canada, Decision Trees, http://chem-eng.utoronto.ca/~datamining/dmc/decision_tree_reg.htm, Zugriff am 05.06.2011
- [24] JDOM, Auf den Seiten von JDOM, <http://www.jdom.org/docs/oracle/jdom-part1.pdf>, Zugriff am 04.06.2011
- [25] Ian H. Witten, Eibe Frank, Len Trigg, Mark Hall, Geoffrey Holmes, and Sally Jo Cunningham, Department of Computer Science, University of Waikato, New Zealand., Weka: Practical Machine Learning Tools and Techniques with Java Implementations
- [26] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer Peter Reutemann, Ian H. Witten, The WEKA Data Mining Software: An Update

- [27] Weka, Weka Manual, Auf den Seiten von Weka,
<http://switch.dl.sourceforge.net/project/weka/documentation/3.6.x/WekaManual-3.6.0.pdf> , Zugriff am 24.06.2011
- [28] Dumas, M. and van der Aalst, W. and Ter Hofstede, A., Process-aware information systems: bridging people and software through process technology, 2005, Wiley-Blackwell
- [29] Martin Kranaret , Von Abfall zur Ressource – 85 Abfallwirtschaftliches Kolloquium der Universität Stuttgart, 2007, Stuttgarter Berichte zur Abfallwirtschaft, Band 91
- [30] Jim Gray, Andreas Reuter, Transaction processing: concepts and techniques, 1993, Morgan Kaufmann Publishers
- [31] F. Sebastiani. Machine learning in automated text categorization. ACM Computing Surveys, 34(1):1–47, 2002.

Anhang

RM Datenbank (komplett)



Erklärung

Hiermit versichere ich, diese Arbeit selbständig verfasst und nur die angegebenen Quellen benutzt zu haben

Alexandru-Tiberiu Pavel