

258321 DKE Projekt

**Gruppe 2**

**Teammitglieder:**

k01607605, Aistleithner Andrea

k01256561, Dusanic Maja

k01356577, Teuchtmann Alexander

k01356229, Tomic Milos

Anforderungsanalyse

Inhaltsverzeichnis

[1. Introduction 2](#_Toc527493644)

[1.1. Purpose 2](#_Toc527493645)

[1.2. Definitions 2](#_Toc527493646)

[1.3. References 2](#_Toc527493647)

[2. System overview 3](#_Toc527493648)

[3. Overall description 3](#_Toc527493649)

[3.1. Design constraints 3](#_Toc527493650)

[3.2. Product functions 3](#_Toc527493651)

[3.3. User characteristics 4](#_Toc527493652)

[3.4. Limitations, assumptions and dependencies 4](#_Toc527493653)

[4. Specific requirements 5](#_Toc527493654)

[4.1. External interface requirements 5](#_Toc527493655)

[4.2. Functional requirements 5](#_Toc527493656)

[4.2.1. Functional Requirement 1 5](#_Toc527493657)

[4.2.2. Functional Requirement 2 5](#_Toc527493658)

[4.2.3. Functional Requirement 3 6](#_Toc527493659)

[4.2.4. Functional Requirement 4 7](#_Toc527493660)

[4.2.5. Functional Requirement 5 8](#_Toc527493661)

[4.2.6. Functional Requirement 6 9](#_Toc527493662)

[4.2.7. Functional Requirement 7 10](#_Toc527493663)

[4.2.8. Functional Requirement 8 11](#_Toc527493664)

[4.2.9. Functional Requirement 9 12](#_Toc527493665)

[4.2.10. Functional Requirement 10 12](#_Toc527493666)

[4.3. Logical database requirements 13](#_Toc527493667)

[4.4. Software system attributes 13](#_Toc527493668)

[4.4.1. Reliability 13](#_Toc527493669)

[4.4.2. Availability 13](#_Toc527493670)

[4.4.3. Security 13](#_Toc527493671)

[4.4.4. Maintainability 13](#_Toc527493672)

[4.4.5. Portability 13](#_Toc527493673)

[4.5. Other requirements 14](#_Toc527493674)

# Introduction

Dieses Dokument soll einen Einblick in die Anforderungen an das Programm, welches im „DKE Praktikum“ behandelt wird, geben. Es stellt die allgemeinen Funktionalitäten, Zielgruppen, sowie speziellere Anforderungen dar.

## Purpose

Ziel dieses Praktikums ist es, eine Performance Evaluierung in Form eines Evaluierungsframeworks zu realisieren. Dieses soll auf das „Contextualized Business Rule Management“ und auf die „Rule Module Inheritance with Modification Restrictions“ angewendet werden. Durch Testläufe sollen dann aussagekräftige Performanceanalysen durchgeführt werden, welche wiederum Daten generieren, die für eine Performancesteigerung weiterverwendet werden können. Diese Daten sollen für weitere Zwecke gespeichert werden.

Zusätzlich sollen Datengeneratoren implementiert werden, welche Datensätze erzeugen, und anschließend vom Evaluierungsframework für die Performanceanalyse verwendet werden.

## Definitions

|  |  |
| --- | --- |
| **Abkürzung** | **Definition** |
| CBR | Contextualized Business Rule Management |
| Vadalog | auf Datalog basierende Wissensgenerierung |
| IEEE | Institute of Electrical and Electronics Engineers |

## References

[1] IEEE Software Engineering Standards Committee, “ISO/IEC/IEEE 29148:2011(E), IEEE Systems and software engineering - Life cycle processes - Requirementsengineering”, January 12, 2011.

# System overview

Das Software System des Programmes besteht aus folgenden Elementen:

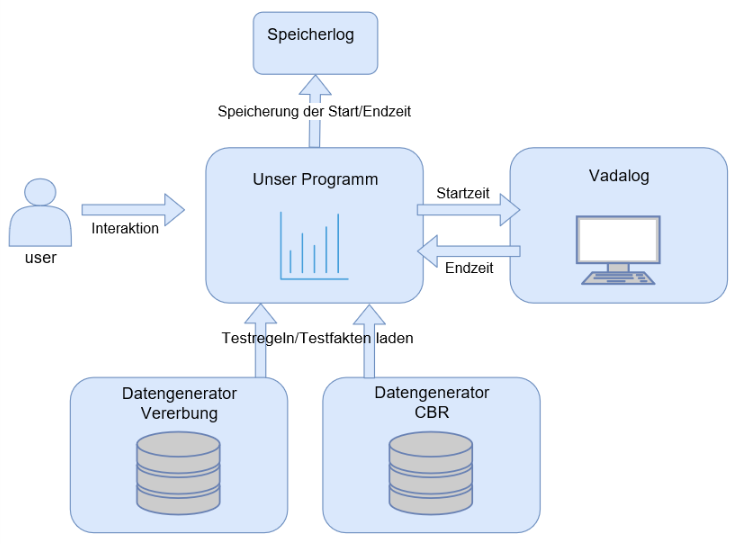


Abbildung 1: Software System Overview

# Overall description

## Design constraints

ToDo

## Product functions

Die allgemeine Funktionalität des Programmes besteht darin, Testdaten für CBR und Inheritance zu generieren um damit verschiedene Testfälle durchführen zu können und somit die Performance der Durchführung messen und evaluieren zu können.

## User characteristics

Insgesamt gibt es drei Arten von Benutzern: „Softwareadministratoren“, „Zuständige für die Geschäftsregeln“ und der spätere Benutzer der Software.

Der Softwareadministrator ist für die Entwicklung und Implementierung der Software und der Schnittstellen verantwortlich. Er ist für den reibungslosen Betrieb des Systems zuständig, sowie für die Sortierung der Geschäftsregeln. Ebenso ist er für die Ausführung der Performanceanalyse, sowie für die Speicherung der Daten, welche auf den Ergebnissen der Analyse basieren, zuständig.

Personen, welche die Geschäftsregeln definieren und laufend, entsprechend der Organisation, aktualisieren, sind ebenso involviert. Diese Regeln werden vom Systemadministrator oder von der jeweilig zuständigen Person in das System eingepflegt.

Der Benutzer der Software wird lediglich mit den für die jeweilige Benutzergruppe vorhergesehenen Informationen konfrontiert. Diese Informationen werden auf Basis der Geschäftsregeln und der Software gefiltert.

## Limitations, assumptions and dependencies

Eine Einschränkung bei der Entwicklung der Performanceanalyse ist, dass jenes grundsätzlich betriebssystemunabhängig arbeiten muss, aber in jedem Fall unter Fedora 28 (Linux) lauffähig sein muss.

# Specific requirements

## External interface requirements

ToDo

## Functional requirements

Im Folgenden werden die Funktionalitäten, welche das Programm haben soll, behandelt.

### Functional Requirement 1

|  |  |
| --- | --- |
| Bezeichnung | Implementierung Datengenerator CBR |
| Beschreibung | Ein Datengenerator zur Erstellung der Regeln und Fakten für auszuführende Testfälle für CBR wird generiert. |
| Anforderungen | need to have |
| Input Werte | * Strings * Integers * Characters |
| Operationsschritte | Durch die Eingabe Werte, werden Test Regeln und Fakten erstellt. |
| Output Werte | generierte Regeln und Fakten |
| Fehlverhalten | bei falscher Eingabe muss die Funktionalität erneut durchgeführt werden. |

### Functional Requirement 2

|  |  |
| --- | --- |
| Bezeichnung | Implementierung Datengenerator Inheritance |
| Beschreibung | Ein Datengenerator zur Erstellung der Regeln und Fakten für auszuführende Testfälle für Inheritance wird generiert. |
| Anforderungen | need to have |
| Input Werte | * Strings * Integers * Characters |
| Operationsschritte | Durch die Eingabe Werte, werden Test Regeln und Fakten erstellt. |
| Output Werte | generierte Regeln und Fakten |
| Fehlverhalten | bei falscher Eingabe muss die Funktionalität erneut durchgeführt werden. |

### Functional Requirement 3

|  |  |
| --- | --- |
| Bezeichnung | Durchführung Performancetest CBR |
| Beschreibung | Es wird ein Test zur Evaluierung der Performance der Durchführung von CBR durchgeführt. |
| Anforderungen | need to have |
| Input Werte | * Anzahl der zu generierenden Regeln * Anzahl der zu generierenden Fakten * Anzahl der Testdurchläufe * Anzahl der Wiederholungen |
| Operationsschritte | Die Anzahl der zu generierenden Regeln wird an den Datengenerator CBR übergeben.  Die Anzahl der zu generierenden Fakten wird an den Datengenerator CBR übergeben.  Die Testdaten werden vom Datengenerator erzeugt.  Die Testdaten werden über die Vadalog Schnittstelle an Vadalog übergeben.  Die Anzahl der Testdurchläufe wird über die Vadalog Schnittstelle an Vadalog übergeben.  Die Anzahl der Wiederholungen wird über die Vadalog Schnittstelle an Vadalog übergeben.  Vadalog führt die Testdaten aus.  Die Startzeit des Tests wird gespeichert.  Die Endzeit des Tests wird gespeichert. |
| Output Werte | * Startzeit des Tests. * Endzeit des Tests. |
| Fehlverhalten |  |

### Functional Requirement 4

|  |  |
| --- | --- |
| Bezeichnung | Durchführung Performancetest Inheritance Single-Inheritance |
| Beschreibung | Es wird ein Test zur Evaluierung der Performance der Durchführung von Inheritance Single-Inheritance durchgeführt. |
| Anforderungen | need to have |
| Input Werte | * Anzahl der zu generierenden Regeln * Anzahl der zu generierenden Fakten * Anzahl der Testdurchläufe * Anzahl der Wiederholungen * Information über Single Inheritance |
| Operationsschritte | Die Anzahl der zu generierenden Regeln wird an den Datengenerator Inheritance übergeben.  Die Anzahl der zu generierenden Fakten wird an den Datengenerator CBR übergeben.  Die Information über Single Inheritance wird an den Datengenerator übergeben.  Die Testdaten werden vom Datengenerator erzeugt.  Die Testdaten werden über die Vadalog Schnittstelle an Vadalog übergeben.  Die Anzahl der Testdurchläufe wird über die Vadalog Schnittstelle an Vadalog übergeben.  Die Anzahl der Wiederholungen wird über die Vadalog Schnittstelle an Vadalog übergeben.  Vadalog führt die Testdaten aus.  Die Startzeit des Tests wird gespeichert.  Die Endzeit des Tests wird gespeichert. |
| Output Werte | * Startzeit des Tests. * Endzeit des Tests. |
| Fehlverhalten |  |

### Functional Requirement 5

|  |  |
| --- | --- |
| Bezeichnung | Durchführung Performancetest Inheritance Multi-Inheritance |
| Beschreibung | Es wird ein Test zur Evaluierung der Performance der Durchführung von Inheritance Multi-Inheritance durchgeführt. |
| Anforderungen | need to have |
| Input Werte | * Anzahl der zu generierenden Regeln * Anzahl der zu generierenden Fakten * Anzahl der Testdurchläufe * Anzahl der Wiederholungen * Information über Multi Inheritance |
| Operationsschritte | Die Anzahl der zu generierenden Regeln wird an den Datengenerator Inheritance übergeben.  Die Anzahl der zu generierenden Fakten wird an den Datengenerator CBR übergeben.  Die Information über Multi Inheritance wird an den Datengenerator übergeben.  Die Testdaten werden vom Datengenerator erzeugt.  Die Testdaten werden über die Vadalog Schnittstelle an Vadalog übergeben.  Die Anzahl der Testdurchläufe wird über die Vadalog Schnittstelle an Vadalog übergeben.  Die Anzahl der Wiederholungen wird über die Vadalog Schnittstelle an Vadalog übergeben.  Vadalog führt die Testdaten aus.  Die Startzeit des Tests wird gespeichert.  Die Endzeit des Tests wird gespeichert. |
| Output Werte | * Startzeit des Tests. * Endzeit des Tests. |
| Fehlverhalten |  |

### Functional Requirement 6

|  |  |
| --- | --- |
| Bezeichnung | Durchführung Performancetest Inheritance mit Modifikation |
| Beschreibung | Es wird ein Test zur Evaluierung der Performance der Durchführung von Inheritance mit Modifikation durchgeführt. |
| Anforderungen | need to have |
| Input Werte | * Anzahl der zu generierenden Regeln * Anzahl der zu generierenden Fakten * Anzahl der Testdurchläufe * Anzahl der Wiederholungen * Information über Modifikation |
| Operationsschritte | Die Anzahl der zu generierenden Regeln wird an den Datengenerator Inheritance übergeben.  Die Anzahl der zu generierenden Fakten wird an den Datengenerator CBR übergeben.  Die Information über die Modifikation wird an den Datengenerator übergeben.  Die Testdaten werden vom Datengenerator erzeugt.  Die Testdaten werden über die Vadalog Schnittstelle an Vadalog übergeben.  Die Anzahl der Testdurchläufe wird über die Vadalog Schnittstelle an Vadalog übergeben.  Die Anzahl der Wiederholungen wird über die Vadalog Schnittstelle an Vadalog übergeben.  Vadalog führt die Testdaten aus.  Die Startzeit des Tests wird gespeichert.  Die Endzeit des Tests wird gespeichert. |
| Output Werte | * Startzeit des Tests. * Endzeit des Tests. |
| Fehlverhalten |  |

### Functional Requirement 7

|  |  |
| --- | --- |
| Bezeichnung | Durchführung Performancetest Inheritance ohne Modifikation |
| Beschreibung | Es wird ein Test zur Evaluierung der Performance der Durchführung von Inheritance ohne Modifikation durchgeführt. |
| Anforderungen | need to have |
| Input Werte | * Anzahl der zu generierenden Regeln * Anzahl der zu generierenden Fakten * Anzahl der Testdurchläufe * Anzahl der Wiederholungen * Information über Modifikation |
| Operationsschritte | Die Anzahl der zu generierenden Regeln wird an den Datengenerator Inheritance übergeben.  Die Anzahl der zu generierenden Fakten wird an den Datengenerator CBR übergeben.  Die Information über die Modifikation wird an den Datengenerator übergeben.  Die Testdaten werden vom Datengenerator erzeugt.  Die Testdaten werden über die Vadalog Schnittstelle an Vadalog übergeben.  Die Anzahl der Testdurchläufe wird über die Vadalog Schnittstelle an Vadalog übergeben.  Die Anzahl der Wiederholungen wird über die Vadalog Schnittstelle an Vadalog übergeben.  Vadalog führt die Testdaten aus.  Die Startzeit des Tests wird gespeichert.  Die Endzeit des Tests wird gespeichert. |
| Output Werte | * Startzeit des Tests. * Endzeit des Tests. |
| Fehlverhalten |  |

### Functional Requirement 8

|  |  |
| --- | --- |
| Bezeichnung | Durchführung Performancetest Inheritance mit Restrictions |
| Beschreibung | Es wird ein Test zur Evaluierung der Performance der Durchführung von Inheritance mit Restrictions durchgeführt. |
| Anforderungen | need to have |
| Input Werte | * Anzahl der zu generierenden Regeln * Anzahl der zu generierenden Fakten * Anzahl der Testdurchläufe * Anzahl der Wiederholungen * Information über Restrictions |
| Operationsschritte | Die Anzahl der zu generierenden Regeln wird an den Datengenerator Inheritance übergeben.  Die Anzahl der zu generierenden Fakten wird an den Datengenerator CBR übergeben.  Die Information über die Restrictions wird an den Datengenerator übergeben.  Die Testdaten werden vom Datengenerator erzeugt.  Die Testdaten werden über die Vadalog Schnittstelle an Vadalog übergeben.  Die Anzahl der Testdurchläufe wird über die Vadalog Schnittstelle an Vadalog übergeben.  Die Anzahl der Wiederholungen wird über die Vadalog Schnittstelle an Vadalog übergeben.  Vadalog führt die Testdaten aus.  Die Startzeit des Tests wird gespeichert.  Die Endzeit des Tests wird gespeichert. |
| Output Werte | * Startzeit des Tests. * Endzeit des Tests. |
| Fehlverhalten |  |

### Functional Requirement 9

|  |  |
| --- | --- |
| Bezeichnung | Durchführung Performancetest Inheritance ohne Restrictions |
| Beschreibung | Es wird ein Test zur Evaluierung der Performance der Durchführung von Inheritance ohne Restrictions durchgeführt. |
| Anforderungen | need to have |
| Input Werte | * Anzahl der zu generierenden Regeln * Anzahl der zu generierenden Fakten * Anzahl der Testdurchläufe * Anzahl der Wiederholungen * Information über Restrictions |
| Operationsschritte | Die Anzahl der zu generierenden Regeln wird an den Datengenerator Inheritance übergeben.  Die Anzahl der zu generierenden Fakten wird an den Datengenerator CBR übergeben.  Die Information über die Restrictions wird an den Datengenerator übergeben.  Die Testdaten werden vom Datengenerator erzeugt.  Die Testdaten werden über die Vadalog Schnittstelle an Vadalog übergeben.  Die Anzahl der Testdurchläufe wird über die Vadalog Schnittstelle an Vadalog übergeben.  Die Anzahl der Wiederholungen wird über die Vadalog Schnittstelle an Vadalog übergeben.  Vadalog führt die Testdaten aus.  Die Startzeit des Tests wird gespeichert.  Die Endzeit des Tests wird gespeichert. |
| Output Werte | * Startzeit des Tests. * Endzeit des Tests. |
| Fehlverhalten |  |

### Functional Requirement 10

|  |  |
| --- | --- |
| Bezeichnung | Speicherung der Evaluierungswerte |
| Beschreibung | Die aus den Tests gewonnen Evaluierungswerte über die Start und die Endzeit werden außerhalb des Programmes gespeichert. |
| Anforderungen | need to have |
| Input Werte | * Startzeit des Tests. * Endzeit des Tests. |
| Operationsschritte | Die Werte werden aus dem Programm exportiert und in einer externen Stelle gespeichert. |
| Output Werte | - |
| Fehlverhalten |  |

## Logical database requirements

ToDo

## Software system attributes

Im Folgenden werden die Attribute Reliability, Availability, Security, Maintainability und Portability behandelt.

### Reliability

Für die Zuverlässigkeit des Systems werden folgende Faktoren benötigt:

* Daten sind korrekt geschrieben (keine Syntax-Fehler)
* Die Semantik der Daten entspricht dem geplanten Ergebnis
* Die Fakten sind korrekt geschrieben (keine Syntax-Fehler)
* Die Semantik der Fakten entspricht dem geplanten Ergebnis
* Die Schnittstelle zum Vadalog ist fehlerfrei

### Availability

Für die Verfügbarkeit des Systems sind folgende Faktoren wichtig:

* Die Benutzung von Checkpoints

### Security

Um die Sicherheit zu gewährleisten werden folgende Punkte benötigt:

* Die Datensicherheit gewährleisten
* Die Datenintegrität bei kritischen Variablen gewährleisten
* Eine beschränkte Kommunikation zwischen einigen Teilen des Programms

### Maintainability

Die Wartbarkeit des Systems wird durch folgende Punkte berücksichtigt:

* Die Benutzung eines simplen Namensschemas
* Die Dokumentation von wichtigen Entscheidungen bezüglich des Codes

### Portability

Für die Erleichterung der Portierung des Systems auf andere Maschinen und/oder Betriebssysteme werden folgende Punkte berücksichtigt:

* Die Benutzung einer portablen Programmiersprache (z.B. Java)
* Die Benutzung eines portablen Compilers

## Other requirements

**Abbildungsverzeichnis**

[Abbildung 1: Software System Overview 3](file:///C:\Users\aistl_000\Documents\GitHub\258321_DKE_PR\Dokumente\Anforderungsanalyse\Vorlage_Anforderungsanalyse.docx#_Toc527381648)