Ostfalia
Hochschule für angewandte Wissenschaften

Wolfenbüttel

Fakultät Informatik

Erkan Garan, 70467533

Praxisbericht

Betreuer:
Prof. Dr. Claus Führner

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere, dass ich alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet habe, und dass die eingereichte Arbeit weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens gewesen ist.

Wolfenbüttel, den 12. Dezember 2022

Kurzfassung

Anwendungsregeln sind eine besondere Form von Anforderungen. Die Erfüllung dieser Regeln ist essentiell für einen sicheren und zuverlässigen Einsatz von Komponenten, die diese Anwendungsregeln voraussetzen. Dieser Praxisbericht wird sich mit der Datenanalyse von Projekten der Siemens Mobility GmbH beschäftigen, in denen Anwendungsregeln bewertet wurden. Ziel der Datenanalyse wird es sein, die Daten in einer für das Anlernen eines unterstützenden KI-Systems zum Bewerten von Anwendungsregeln geeigneten Form aufzubereiten. Dafür werden Daten von Projekten gesammelt, welche Anwendungsregeln bewertet haben. Daraufhin muss ermittelt werden, welche Attribute für das Anlernen eines KI-Systems relevant sein könnten. Mit diesen Erkenntnissen sollen die Daten dann aufbereitet werden. Das Ergebnis der Datenaufbereitung wird dann in diesem Praxisbericht dargestellt.

Abstract

Application rules are a special form of requirements. The fulfillment of these rules is essential for the safe and reliable use of components that require these application rules. This report will deal with the data analysis of projects of the Siemens Mobility GmbH, which have evaluated application rules. The goal of the data analysis will be to prepare the data in a form suitable for training a supportive AI system, which will be used to evaluate application rules. For this purpose, data will be collected from projects that have evaluated application rules. It will then be necessary to determine which attributes will be relevant for training the AI system. With these findings, the data can then be prepared. The result of the data preparation is then presented in this report.

Inhaltsverzeichnis

Abbildungsverzeichnis

Quellcodeverzeichnis

Abkürzungsverzeichnis

CRISP-DM Cross Industry Standard Process for Data Mining

DOORS IBM Rational DOORS

DXL DOORS Extension Language

1 Einleitung

Schwerpunkt dieses Praxisberichts wird die Analyse der vorhandenen Daten über Projekte der Siemens Mobility GmbH sein, die Anwendungsregeln bewertet haben. Zu Beginn wird zunächst definiert, was Anwendungsregeln und Anforderungen sind und was das Anforderungsmanagement-Tool IBM Rational Doors ist. Danach werden zwei Phasen eines Data-Mining Modells auf den vorhandenen Daten angewandt. Zum Schluss soll eine .csv-Datei erstellt werden, welche einen Datensatz mit allen bewerteten Anwendungsregeln beinhalten soll.

1.1 Motivation

Der Grund für die Analyse und Aufbereitung der Daten zu bewerteten Anwendungsregeln ist das anschließende Anlernen eines unterstützenden KI-Systems. Jenes KI-System soll in der Lage sein, Systems Managern beim Bewerten von Anwendungsregeln zu unterstützen, indem es dem Systems Manager Vorschläge zur Bewertung jeder Anwendungsregeln basierend auf vorherigen Bewertungen zu eben jener Anwendungsregel gibt.

1.2 Zielsetzung

Ziel des Praxisprojektes ist es, dass aus der Menge an Daten von bereits abgeschlossenen und noch laufenden Projekten der Siemens Mobility GmbH die Daten herausgefiltert werden, die zum Anlernen des KI-Systems geeignet sind. Zudem sollen die Daten bereits so aufbereitet werden, dass ausschließlich für das Anlernen relevante Daten vorhanden sind. Mithilfe dieser Daten soll dann das KI-System angelernt werden, was jedoch nicht mehr Teil dieser Arbeit ist, sondern im Anschluss in einer

Bachelorarbeit stattfinden soll, welche auf dieses Praxisprojekt und den dazugehörigen Praxisbericht aufbauen soll.

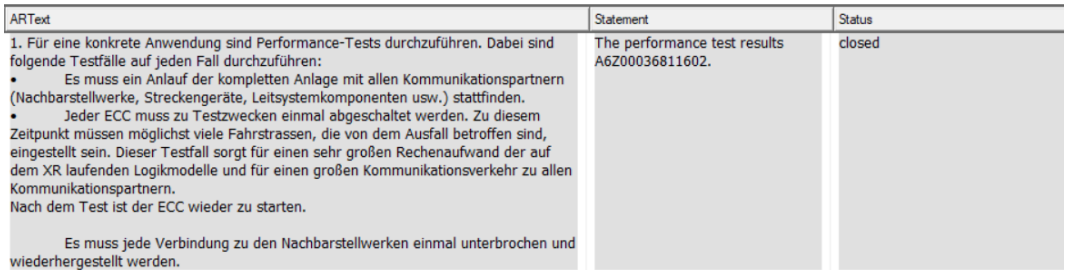
2 Bewerten von Anwendungsregeln

Anwendungsregeln sind Anforderungen von Komponenten oder Teilsystemen, die der Kunde oder das Projekt, das die jeweilige Komponente oder das Teilsystem einsetzt, zu beachten hat. Diese Regeln müssen erfüllt werden, um einen sicheren und zuverlässigen Einsatz jeder Komponente im System zu gewährleisten. Dabei können Anwendungsregeln auch sicherheitsrelevant sein. Ist dies der Fall, werden sie auch sicherheitsrelevante Anwendungsregeln genannt [3, S.9]. Eine Anforderung wird nach der IEEE entweder definiert als eine Bedingung oder eine Fähigkeit, die ein Benutzer benötigt, um ein Problem zu lösen oder ein Ziel zu erreichen oder als eine Bedingung oder Fähigkeit, die ein System oder eine Systemkomponente erfüllen oder besitzen muss, um einen Vertrag, eine Norm, eine Spezifikation oder ein anderes formal vorgeschriebenes Dokument zu erfüllen [4, S.62]. Somit stellen Anwendungsregeln eine besondere Form von Anforderungen dar. Eine sicherheitsrelevante Anwendungsregel für ein Zugbeeinflussungssystem könnte wie folgt aussehen:

> *Beispiel Anwendungsregel* „In order to prevent accidents normally covered by Train Control System, the driver shall assume full safety responsibility for the operation of a train if he / she activates a cab on a vehicle with cut-off switch in „ATP off“ position. If the cut-off switch on a cab is in „ATP off“ position, the Rolling Stock cut-off-circuitry bypasses all safety related outputs of the Train Control System.“ [5, S.7].

Nutzt ein Projekt nun ein anderes System oder eine andere Komponente, muss der Systems Manager, also derjenige im Projekt, der sich um die Anforderungs-

bearbeitung kümmert, die Anwendungsregeln des Systems oder der Komponente im Anforderungsmanagement-Tool DOORS importieren lassen und diese bewerten. Die Anwendungsregeln von bereits vorhandenen Systemen und Komponenten liegen zentral gespeichert in DOORS. Bei dieser Bewertung muss der Systems Manager prüfen, ob die importierten Anwendungsregeln im Projekt berücksichtigt werden müssen oder nicht und wie die Einhaltung der Anwendungsregeln sichergestellt wird. Ebenfalls muss der Systems Manager ein Statement abgeben, wie er zu dieser Bewertung gekommen ist. Eine Anwendungsregel und ihre Bewertung können der Abbildung 2.1 entnommen werden.



ARText	Statement	Status
1. Für eine konkrete Anwendung sind Performance-Tests durchzuführen. Dabei sind folgende Testfälle auf jeden Fall durchzuführen: • Es muss ein Anlauf der kompletten Anlage mit allen Kommunikationspartnern (Nachbarstellwerke, Streckengeräte, Leitsystemkomponenten usw.) stattfinden. • Jeder ECC muss zu Testzwecken einmal abgeschaltet werden. Zu diesem Zeitpunkt müssen möglichst viele Fahrstrassen, die von dem Ausfall betroffen sind, eingestellt sein. Dieser Testfall sorgt für einen sehr großen Rechenaufwand der auf dem XR laufenden Logikmodelle und für einen großen Kommunikationsverkehr zu allen Kommunikationspartnern. Nach dem Test ist der ECC wieder zu starten. Es muss jede Verbindung zu den Nachbarstellwerken einmal unterbrochen und wiederhergestellt werden.	The performance test results A6Z00036811602.	closed

Abbildung 2.1: Beispiel einer bewerteten Anwendungsregel in DOORS

3 IBM Rational DOORS

Rational DOORS ist ein Anforderungsmanagement-Tool des US-amerikanischen IT-Unternehmens IBM. Das Akronym DOORS steht für Dynamic Object-Oriented Requirements System [6]. Dieses Tool wird zur Dokumentation und Verwaltung von Anforderungen genutzt. Die DOORS Datenbank ist dabei hierarchisch aufgebaut und besteht aus drei verschiedenen Elementen, nämlich den Projekten, den Ordnern und den Modulen. Dieser Aufbau ist in Abbildung 3.1 dargestellt. Projekte und Ordner werden dafür genutzt, um Daten innerhalb der DOORS Datenbank zu organisieren und zu strukturieren. Projekte sind in der GUI erkennbar durch ein blaues Ordner-Symbol, während Ordner durch ein gelbes Ordner-Symbol gekennzeichnet werden. Dies kann ebenfalls der Abbildung 3.1 entnommen werden. Projekte und Ordner unterscheiden sich vor allem dadurch, dass die Namen von Projekten in der gesamten Datenbank eindeutig sein müssen. Die Position aller Daten innerhalb eines Projekts ist durch den Pfad, welcher mit dem Projekt beginnt, festgelegt [7]. Die Funktionsweise von Ordnern hingegen, ist vergleichbar zu den Ordnern im Windows Explorer.

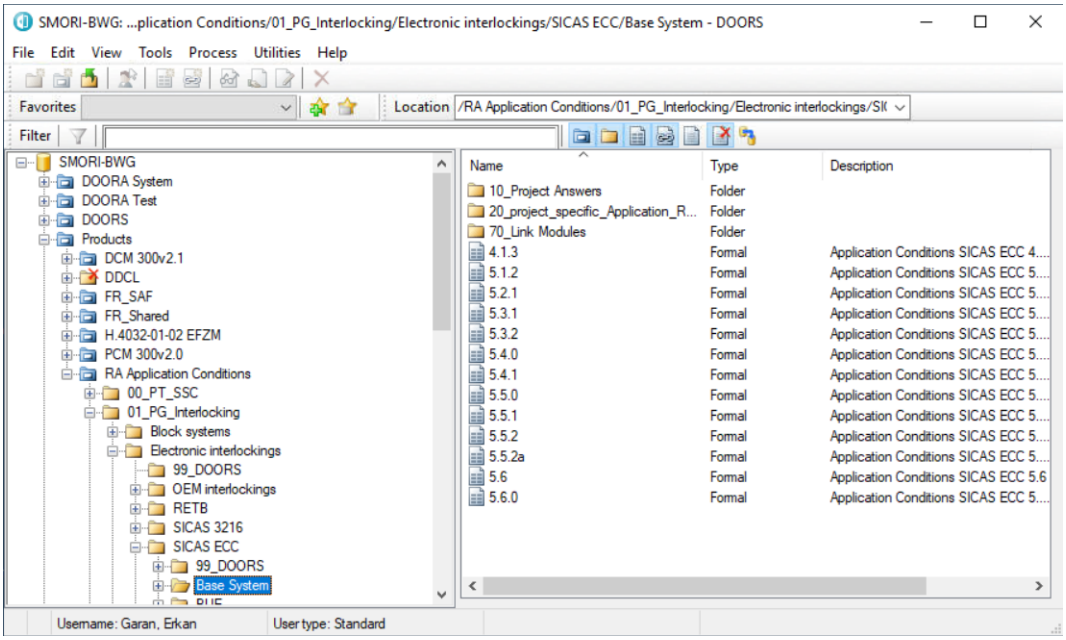


Abbildung 3.1: GUI der DOORS Anwendung

Die Module in DOORS sind die Elemente, welche die Anforderungen in einer tabellarischen Form beinhalten. Abbildung 3.2 zeigt ein geöffnetes Modul in DOORS. Dort ist zu erkennen, dass die einzelnen Spalten die Attribute darstellen, die einzelnen Zeilen repräsentieren dabei die Objekte. Zudem hat der User die Möglichkeit oben links neben dem Feld „View“ eine View auszuwählen. Mithilfe einer View ist der User in der Lage, durch Filter, Sortierungen, Ein- bzw. Ausblenden von Spalten u.Ä. sich nur das anzeigen zu lassen, was für ihn aktuell relevant ist. Rechts neben dem Text des Objekts sind zwei verschiedene Pfeile zu erkennen. Diese Pfeile zeigen an, dass Links, also Verbindungen von oder zu diesem Objekt bestehen. Links sind essentiell für das Requirements Tracing, also für die Verfolgung von Anforderungen. Dadurch wird dokumentiert, von wo eine Anforderung herkommt oder wo sie ebenfalls genutzt wird. Der rote Pfeil mit der Spitze nach rechts ist dabei ein Out-Link, also eine Verbindung von dem Objekt zu einem anderen Objekt. Der gelbe Pfeil mit der Spitze in die entgegengesetzte Richtung symbolisiert einen In-Link, also eine Verbindung von einem anderen Objekt zu diesem Objekt. Eine weitere Form von Modulen sind die Link-Module. Diese werden dafür verwendet, um Informationen über Links zu speichern.

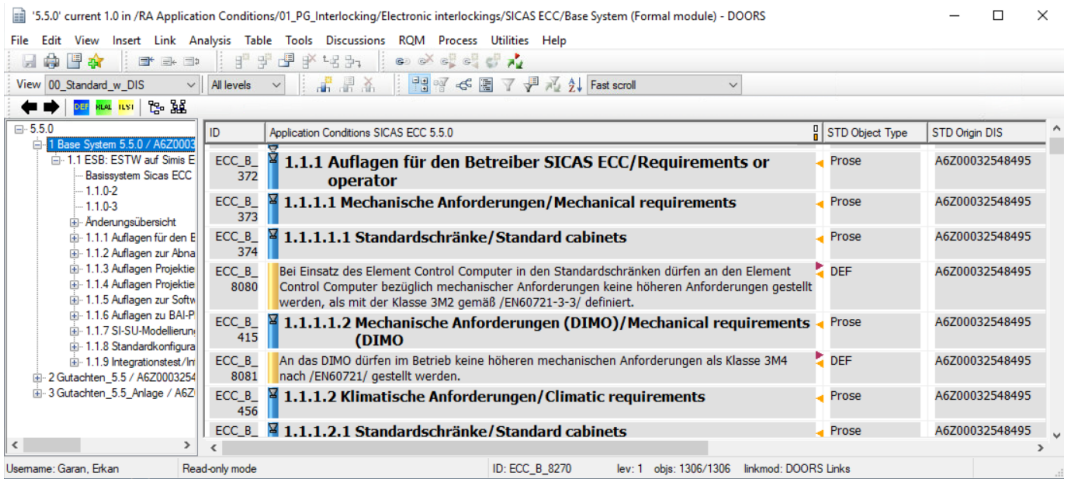


Abbildung 3.2: Modul in DOORS

Neben der Arbeit mittels der grafischen Benutzeroberfläche bietet die DOORS Extension Language (DXL) eine weitere Möglichkeit, um mit DOORS zu arbeiten. DXL ist dabei eine Scripting-Sprache, deren Syntax der von C und C++ ähnelt und dafür genutzt werden kann, um Skripte zu erstellen. Mit diesen Skripten können beispielsweise Module oder Ordner erstellt werden, sowie Objekte angelegt und editiert werden [8].

4 Analyse vorhandener Daten

Ein Modell, das beim Data Mining genutzt werden kann, ist das Cross Industry Standard Process for Data Mining (CRISP-DM). Diese Modell definiert das Data Mining als Prozess mit sechs verschiedenen Phasen und kann der Abbildung 4.1 entnommen werden. Dabei dürfen diese Phasen nicht als starr aufeinander folgend betrachtet werden. Denn auch Erkenntnisse aus nachfolgende Phasen können vorherige Phasen beeinflussen, was durch die Pfeile zwischen den Phasen in der Abbildung 4.1 deutlich wird. In der ersten Phase, dem Business Understanding, liegt der Schwerpunkt auf dem Verständnis der Projektziele aus der Geschäftsperspektive. Mit dem erlangten Wissen daraus, kann eine Data-Mining Problemstellung definiert werden, wie dies in Kapitel 1.1 erfolgt ist. Die nächste Phase ist das Data Understanding. Diese Phase beschäftigt sich mit der ersten Datenerhebung und dem Aufbau eines Datenverständnisses. Zudem wird in dieser Phase versucht, mögliche Probleme bei der Qualität der Daten zu identifizieren. Nach der Phase des Data Understandings folgt die Phase der Data Preparation. Während dieser Phase ist das Ziel, die Rohdaten so aufzubereiten, dass ein Datensatz dabei erstellt wird, der für eine Modellierung geeignet ist. Die nächsten drei Phasen beschäftigen sich mit der Modellierung, der Evaluierung von Modellen und dem Einsatz eines Modells und gehen somit über die Aufgabenstellung des Praxisprojekt hinaus und werden deshalb im Folgenden nicht berücksichtigt. [1, S.5-7]

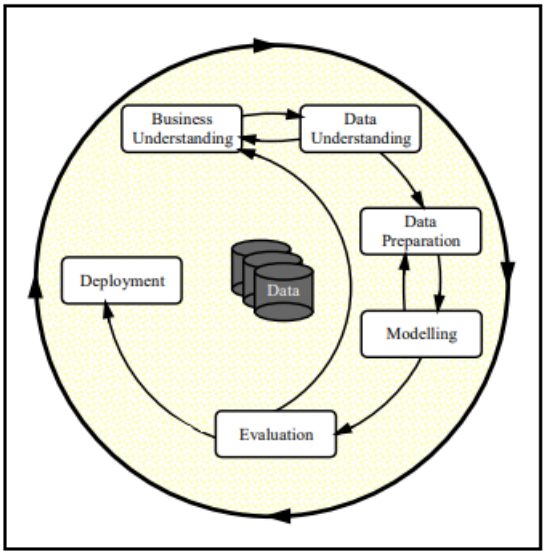


Abbildung 4.1: CRISP-DM Phasen [1, S.5]

4.1 Data Understanding

In der Baumansicht auf der linken Seite der Abbildung 3.1 befindet sich ein Projekt mit dem Namen „RA Application Conditions“. In diesem Projekt befinden sich Module, welche Anwendungsregeln von diversen Komponenten beinhalten. Dafür wurde ein Skript in DXL geschrieben, welches über alle Module innerhalb des Projekts iteriert. Wenn das aktuelle Element ein Modul ist, muss dieses geöffnet werden und falls dieses nicht NULL ist, muss über jedes Objekt in diesem Modul iteriert werden. Bei jedem Objekt muss daraufhin über alle In-Links iteriert werden. Über diese In-Links ist es möglich, das Quellmodul zu bestimmen, also das Modul, von dem der Link kommt. Diese Quellmodule sind potentiell relevante Module für das Anlernen des KI-Systems. Nach dem Process Manual für Anwendungsregeln der Siemens Mobility GmbH [3, S.32] muss das Link-Modul des Links den Namen „reference_of“ tragen. Bei älteren Projekten jedoch wurde als Name „42_reference_of“ genutzt. Zudem kann der Fall auftreten, dass Quellmodule in einer Sandbox eines Mitarbeiters liegen. Diese Module dürfen nicht berücksichtigt werden, da sie kein Teil von Projekten der Siemens Mobility GmbH sind. Alle Quellmodule, die den Anforderungen entsprechen, werden in eine Skiplist eingefügt. Skiplisten sind eine Datenstruktur, welche aus einem Key-Value-Paar bestehen.

```
Folder f = folder("/RA Application Conditions");
void searchInLinks(Folder f){
    for it in f do{
        if (type(it) == "Folder"){
            searchInlinks(folder(it));
        }
        if (type(it) == "Formal"){
            m = read(fullName it, false);
            if (m != null){
                for o in entire m do{
                    for lr in all (o <- "*") do {
                        if (name module lr == "42_reference_of"
                        || name module lr == "reference_of") {
                            srcModRef = source lr;
                            iOffset = null;
                            iLength = null;
                            if(!(findPlainText(fullName srcModRef,
                            "Sandbox", iOffset, iLength, false))){
                                put(slModules, fullName srcModRef,
                                fullName srcModRef);
                                // ...
```

Quellcode 4.1.1: Iterieren über alle Module von RA Application Conditions

Nach dem Iterieren aller Module im Projekt RA Application Conditions, lassen sich zurzeit 115 Module aus 63 Projekten finden, welche einen Link zu Anwendungsregeln besitzen und diese somit importiert haben. Diese Projekte mit ihren Modulen sind potentielle Kandidaten für einen Datensatz zum Anlernen des KI-Systems. Als nächsten Schritt müssen diese Module genauer betrachtet werden. Es muss geprüft werden, welche Attribute für das Bewerten von Anwendungsregeln relevant sind. Der Quellcode 4.1.2 zeigt eine Möglichkeit, wie alle Attribute eines Moduls ausgegeben werden können. Nach dem Process Manual für Anwendungsregeln müssen Projekte, welche Anwendungsregeln importiert haben, die Attribute REQ Status und REQ Statement beinhalten. Das Attribut REQ Status vom Typ einer single-enumeration beinhaltet dabei die Bewertung der Anwendungsregel. Die Begründung dafür lässt sich im Attribut REQ Statement in Textform finden[3, S.14].

```
1 string modAttrName
2 for modAttrName in attributes (current Module) do
3     print modAttrName "\n"
```

Quellcode 4.1.2: Alle Attribute eines Moduls ausgeben [2]

Die folgende Aufzählung beinhaltet eine Auswahl von Attributen eines beispielhaft gewählten Moduls.

- Absolute Number
- ALinkRef
- Compare
- Last Modified On
- Object Heading
- Object Short Text
- Object Text
- OLE
- OLEIconic
- Picture
- PictureName
- PictureNum
- REQ Comment_int
- REQ Domain
- REQ Statement
- REQ Status
- REQ Type
- RTF Pict
- STD Explanation
- STD Identifier
- STD Title

Für das Bewerten der Anwendungsregel ist der Object Text, also die Anwendungsregel in Textform, sowie die Attribute REQ Statement und REQ Status relevant, da das KI-System später Vorschläge für diese beiden Attribute liefern soll. Die Analyse von Modulen von Projekten, welche nach 2019 durchgeführt wurden, hat gezeigt, dass dort das Attribut REQ Status den Namen REQ Progress trägt. Darauf

muss bei dem Sammeln der Daten und der späteren Aufbereitung dieser geachtet werden. Zudem wurde dabei deutlich, dass einige Module weder REQ Status bzw. REQ Progress noch REQ Statement als Attribut besitzen. Mögliche Erklärungsansätze dafür wären, dass die Projektausführung von den Vorgaben des Process Manuals abweichen, dass die Projekte sich noch in einem frühen Stadium befinden und deshalb die Anwendungsregeln nicht bewertet wurden oder dass die Anwendungsregeln lediglich importiert, aber nicht bewertet wurden.

Außerdem muss beachtet werden, dass ein Projekt mittels eines Copy-Update-Tools die Anwendungsregeln auf eine neue Version updaten kann. Die alten Versionen der Anwendungsregeln werden dann unter ein Objekt verschoben, welches die Überschrift „Old Version Objects, to be reviewed and deleted: “ hat. Diese Anwendungsregeln dürfen nicht verwendet werden, da diese veraltet sein könnten.

4.2 Data Preparation

Mit den Erkenntnissen aus der vorherigen Phase beginnt nun die Aufbereitung der Daten. Jede Komponente bzw. jedes System hat im Project RA Application Conditions einen Ordner „10_Project Answers“. Nutzt ein Projekt nun beispielsweise das Achszähler-System AzS350U und hat die Anwendungsregeln des Systems bewertet, dann müssen diese bewerteten Anwendungsregeln in ein neues Modul in dem „10_Project Answers“ Ordner des Achszähler-Systems geschrieben werden [3, S.21]. Mit den Modulen, die die bewerteten Anwendungsregeln der einzelnen Komponenten und Systeme beinhalten, kann dann das KI-System angelernt werden.

Um das zu erreichen, müssen zunächst die 115 Module genauer untersucht werden. Dabei muss geprüft werden, ob diese Module die Anwendungsregeln so bewertet haben, wie es das Process Manual vorgibt. Dafür muss die Iteration aus 4.1.1 erweitert werden. Es muss zusätzlich geprüft werden, ob das Modul die Attribute REQ Status bzw. REQ Progress besitzt und ob diese auch ausgefüllt sind.

```
if(hasAttribute("REQ Status", mod)
&& hasAttribute("REQ Statement", mod)){
    szProgress = o."REQ Status""";
    szStatement = o."REQ Statement";
    if(length(szProgress)>1 && length(szStatement)>1){
        put(slResults, fullName mCheck, fullName mCheck)
    }
}else if (hasAttribute("REQ Progress", mod)
&& hasAttribute("REQ Statement", mod)){
    szProgress = o."REQ Progress""";
    szStatement = o."REQ Statement";
    if(length(szProgress)>1 && length(szStatement)>1){
        put(slResults, fullName mCheck, fullName mCheck)
        // ...
```

Quellcode 4.2.1: Suche nach bewerteten Anwendungsregeln

Übrig bleiben danach 41 Module aus 34 Projekten, welche nicht nur die Anwendungsregeln importiert, sondern auch nach den Vorgaben des Process Manuals bewertet haben. Als Zwischenschritt wurde daraufhin für jedes Projekt ein Modul in meiner Sandbox erstellt. In diesem Modul wurden dann alle bewerteten Anwendungsregeln zusammengestellt. Dabei musste darauf geachtet werden, dass ausschließlich die Anwendungsregeln in das Modul geschrieben werden und keine Überschriften oder Artefakte. Zudem durften nur die relevanten Attribute in dem

neuen Modul stehen. Alle weiteren Attribute dürfen nicht hinzugefügt werden, da diese für das Bewerten von Anwendungsregeln irrelevant sind und die Module und somit den schlussendlichen Datensatz, welcher zum Trainieren des KI-Systems verwendet werden soll, unnötig groß machen würden. Zudem muss, wie in Kapitel 4.1 beschrieben, überprüft werden, ob sich die Anwendungsregel unter dem Punkt „Old Version Objects, to be reviewed and deleted: “ befindet. Dafür muss über alle übergeordneten Objekte eines Objekts iteriert werden und geprüft werden, ob die Überschrift eines Objekts „Old Version Objects, to be reviewed and deleted: “ ist. Der Quellcode 4.2.2 zeigt eine Möglichkeit, bei der eine Boolean-Variable auf true gesetzt wird, falls ein Objekt veraltet ist. Diese Objekte werden dann nicht in das neue Modul geschrieben.

```
for obj in entire mod do{
    oParent = parent(obj)
    bOldVersion = false;
    while (oParent != null){
        if (oParent."Object Heading""" ==  "Old Version Objects,
   to be reviewed and deleted:"){
            bOldVersion = true;
        }
        oParent = parent(oParent)
    }
```

Quellcode 4.2.2: Prüfen, ob Objekt veraltet ist

Abbildung 4.2 zeigt, wie ein Modul für ein Projekt dann aussieht. Erkennbar ist, dass dort lediglich relevanten Attribute gespeichert werden. Zudem wird ein Out-Link auf die Anwendungsregel unter RA Application Conditions gesetzt, damit nachverfolgt werden kann, von wo die Anwendungsregel ursprünglich stammt. Des Weiteren besteht der Name des Moduls aus dem Namen des Projekts, sowie der Endung _Solutions. Dadurch wird sichergestellt, dass die Information darüber, von welchem Projekt die bewertete Anwendungsregel stammt, erhalten bleibt.

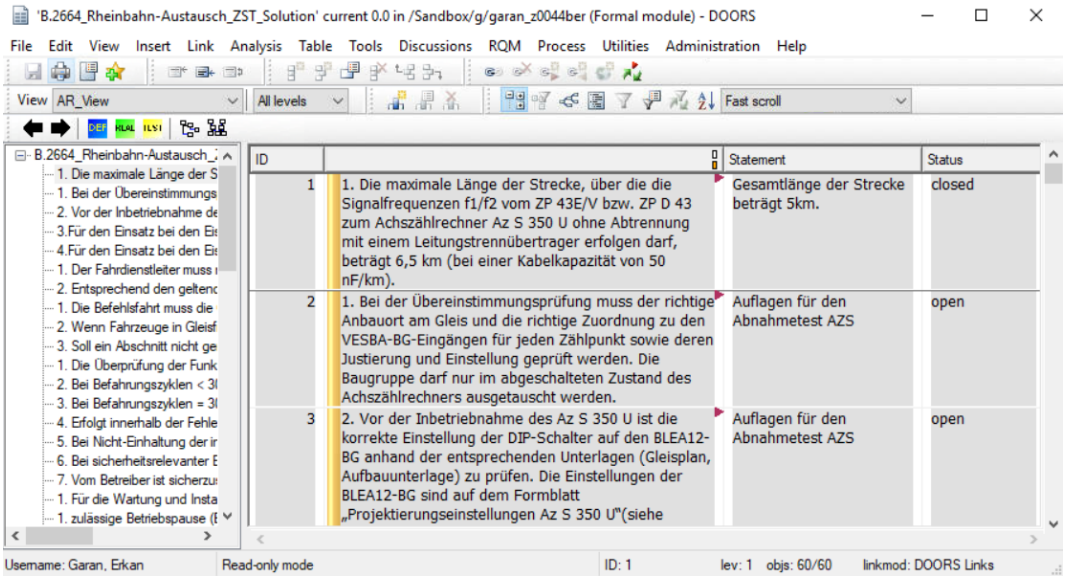


Abbildung 4.2: Modul eines Projekts

Im nächsten Schritt müssen diese Solution-Module der einzelnen Projekte noch weiter aufgeteilt werden. Durch den Out-Link jedes Objekts kann geprüft werden, von welcher Komponente oder System die Anwendungsregel stammt. Daraufhin müssen alle Anwendungsregeln eines Projekts, die von der selben Komponente oder vom selben System stammen, in ein eigenes Modul ausgegliedert werden, welches dann in den Project Answers Ordner der Komponente oder des Systems verschoben wird. Dafür wird über alle Objekte im Solution-Modul eines Projekts iteriert. Jedes Objekt wird dann in ein Modul kopiert, dessen Name sich aus dem Projektnamen, sowie des Namens der Komponente oder des Systems zusammensetzt. Letzterer Name wird dabei aus dem Out-Link übernommen. Diese Module müssen dann zum Schluss in den jeweilegen Project Answers Ordner verschoben werden.

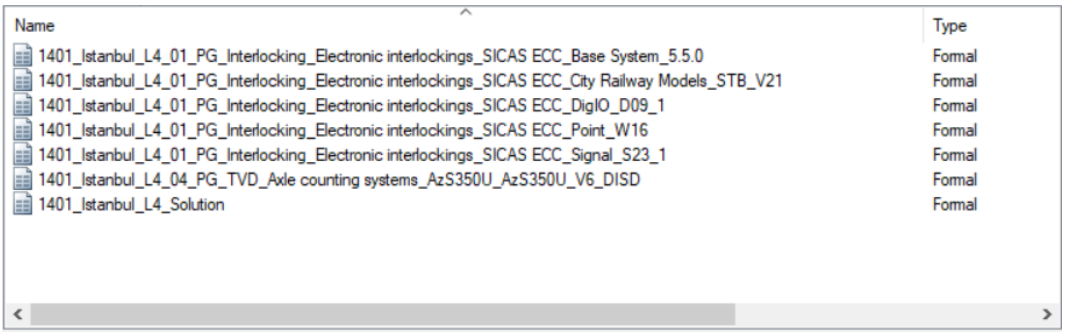


Abbildung 4.3: Aufgeteiltes Solution-Modul

Bevor die Module verschoben werden, wird vorher der Datensatz zum Anlernen des KI-Systems im .csv-Format erstellt. Dieser Schritt wird eingeschoben, da zu diesem Zeitpunkt die Module zentral in meiner Sandbox liegen und es somit einfacher ist, über all diese Module zu iterieren. Dabei wird wieder über jedes Objekt von jedem Modul iteriert und dabei der Objekt Text, der Status, sowie das Statement in eine Variable vom Typ String geschrieben. Bei dem Objekt Text und bei dem Statement müssen alle Kommata entfernt werden, da diese sonst zu Problemen bei der .csv-Datei führen würden. Dafür wurde eine Methode erstellt, die alle Kommata eines Textes entfernt. Anschließend werden die drei Variablen an eine String-Variable konkateniert, welche am Ende jedes Moduls in die .csv-Datei geschrieben wird. Für das Schreiben in diese Datei musste ebenfalls eine weitere Methode geschrieben werden. Der String, welcher in die Datei geschrieben wird, hat dann die Form:

"Anwendungsregel","Status","Statement"
"Anwendungsregel","Status","Statement"
...

```
// ...
m = read(fullName(it), false)
for o in entire m do{
    szText = cleanText(o."Object Text""")
    szStatus = o."Status"
    szStatment = cleanText(o."Statement""")
    szData = szData "\"" szText "\",\""szStatus"\",\""szStatment"
    \"\n"
}
writeCSV(szData)
// ...
```

Quellcode 4.2.3: Daten in eine .csv-Datei schreiben

Nun wird eine .csv-Datei mit dem finalen Datensatz erstellt, welcher später für das Anlernen eines unterstützenden KI-Systems zum Bewerten von Anwendungsregeln verwendet werden kann. Dieser Datensatz umfasst dabei 195.518 bewertete Anwendungsregeln aus Projekten der Siemens Mobility GmbH. Zum Schluss können nun alle Module in ihre entsprechenden Project Answers Ordner verschoben werden.

5 Schluss

Das Ziel dieses Praxisprojekts war es, einen für das Anlernen eines KI-Systems geeigneten Datenpool zu erstellen. Nach dem Model CRISP-DM wurden dabei die Projekte, welche Anwendungsregeln genutzt haben, untersucht und analysiert. Zu Beginn wurde zunächst nach potentiell relevanten Projekten in der DOORS-Datenbank gesucht. Realisiert wurde diese Suche mit Skripten in der Sprache DXL. Diese Projekte und ihre dazugehörigen Module wurden daraufhin genauer untersucht. Dabei wurde geprüft, welche Attribute relevant sein könnten und welche nicht. Das Process Manual wurde dabei genutzt, um ein besseres Verständnis der einzelnen Attribute zu erlangen und die vorgeschriebene Vorgehensweise beim Bewerten von Anwendungsregeln nachzuvollziehen. An dieser Stelle fiel auf, dass einige Projekte dabei ihre importierten Anwendungsregeln nicht oder noch nicht bewertet hatten. Deshalb mussten jene Projekte, welche keine Bewertungen zu ihren Anwendungsregeln besaßen, aussortiert werden, da diese keinen Mehrwert für den zu erstellenden Datenpool boten. Im nächsten Schritt mussten neue Module für jedes Projekt erstellt werden. Die neu erstellten Module durften dabei nur für das Anlernen des KI-Systems relevante Attribute besitzen. Mithilfe dieser neu erstellten Module für jedes Projekt konnten dann alle Anwendungsregeln in den Modulen mit ihren Bewertungen in eine .csv-Datei geschrieben werden. Abschließend wurden die Module jedes Projekts aufgeteilt. Dafür wurde für jede Komponente oder jedes System, dessen Anwendungsregeln in einem Projekt bewertet wurden, ein eigenes Modul erstellt. Abschließend wurden diese aufgeteilten Module dann in ihren jeweiligen Project Answers Ordner verschoben, wie es das Process Manual für Anwendungsregeln vorgibt [3, S.32].

Durch dieses Praxisprojekt wurde ein Datenpool über 195.518 bewertete Anwendungsregeln erstellt, welcher für das Anlernen eines unterstützenden KI-Systems zum Bewerten von Anwendungsregeln genutzt werden kann. Dieser Bericht be-

trachtet dabei das Erstellen eines KI-Systems nicht und bietet deshalb die Möglichkeit, als Grundlage für eine weitere Arbeit zu dienen, die sich mit der Erstellung und dem Anlernen eines KI-Systems beschäftigt.

Literaturverzeichnis

[1] R. Wirth and J. Hipp, "CRISP-DM: Towards a standard process model for data mining," *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, vol. 1, 2000.

[2] IBM, *DXL Reference Manual*.

[3] Siemens Mobility GmbH, *RM Process Manual - Application Rules*, 2016.

[4] IEEE(The Institute of Electrical and Electronics Engineers), *IEEE Standard Glossary of Software Engineering Terminology*, 1990.

[5] Siemens Mobility GmbH, *Safety-Related Application Rules*, 2016.

[6] "Übersicht über Rational DOORS," https://www.ibm.com/docs/de/ermd/9.6.1?topic=overview-rational-doors, accessed: 2022-11-25.

[7] "Daten in Projekten und Ordnern organisieren," https://www.ibm.com/docs/de/ermd/9.6.0?topic=requirements-organizing-data-in-projects-folders, accessed: 2022-11-25.

[8] "Rational DOORS mit DXL erweitern," https://www.ibm.com/docs/de/ermd/9.6.1?topic=function-extending-rational-doors-dxl, accessed: 2022-11-28.