

Pflichtenheft



Schaltsysteme-Arbeitsblätter im Netz *Softwareprojekt 2018*

Dr.-Ing. Heinz-Dietrich Wuttke
René Hutschenreuter

David Sukiennik

Alexander Beyer
Lars Hinneburg
Marcel Burkhardt
Sarah Löcklin

Eike Krieg
Linda Oppermann
Peter Klein
Stephan Enseleit

Inhaltsverzeichnis

1	Einleitung.....	3
1.1	Musskriterien	3
1.2	Wunschkriterien	4
1.3	Mögliche Erweiterungen	4
1.4	Abgrenzungskriterien	4
2	Produkteinsatz	5
2.1	Zielgruppen	5
2.2	Betriebsbedingungen	5
3	Produktumgebung	5
4	Produktfunktionen	6
5	Produktdaten	7
6	Produktleistungen	8
7	Vorgeschlagenes System	8
7.1	Benutzeroberfläche.....	8
7.1.1	Allgemeines zur graphischen Benutzeroberfläche	8
7.1.2	Details zur Benutzeroberfläche	9
7.2	funktionale Anforderungen.....	9
7.3	Nichtfunktionale Anforderungen	14
7.4	Qualitätszielbestimmungen	16
7.5	globale Testfälle und -szenarien	16
7.6	Anwendungsfälle.....	16
8	Glossar.....	23

1 Einleitung

Das Ziel des Projekts SANE: „Schaltssysteme Arbeitsblätter im Netz“ liegt in der Begleitung und Unterstützung des Lehr- und Praktikumsbetriebs der Fächer am Fachgebiet Integrierte Kommunikationssysteme. Hierfür wird die Webapplikation SANE entwickelt, welche wichtige Grundelemente zur Darstellung boolescher Funktionen wie Wertetabellen, Mengendiagramme, boolesche Gleichungen, Funktionsindizes, boolesche Algebra, Karnaugh-Veitch-Diagramm, Hasards, Quine-McCluskey, ROM, PLA, GAL und evtl. g-Parameter zur Verfügung stellt.

1.1 Musskriterien

Dem Nutzer stehen folgende Views zur Verfügung:

- eine skalierbare Wertetabelle mit konfigurierbar vielen Eingangs- und Ausgangsvariablen,
- ein Karnaugh-Veitch-Diagramm, welches bis zu 6 Eingangsvariablen unterstützt und automatisch die Blöcke für die Minimierung bilden kann,
- ein Venn-Diagramm,
- eine Ansicht der zentralen booleschen Funktion in KDNF, KKNF, KNF und DNF,
- die Darstellung des QMC-Algorithmus mit veranschaulichten Zwischenschritten der Prozedur,
- eine Erkennung von Funktions-Hasards in einem KV-Diagramm und
- die Anzeige und Eingabe von Funktionsindizes als Hexadezimalzahl.

Des Weiteren stehen dem Nutzer folgende Einstellungsmöglichkeiten zur Verfügung:

- eine Sprachauswahl zwischen Englisch und Deutsch,
- ein heller und ein dunkler Farbmodus,
- ein wählbarer Zeichensatz für die boolesche Ausdrucksalgebra,
- eine Reset-Funktion,
- die Übergabe einer Startkonfiguration und
- eine Import-/Export-Funktion.

1.2 Wunschkriterien

Folgende Kriterien sind Wunschkriterien und werden erst nach Implementierung der Musskriterien bearbeitet:

- ein View setzt programmierbare Strukturen, wie ROM, PLA und GAL, angelehnt an die Schematik der Lehrveranstaltungen um,
- Installation von SANE auf Smartphones und Tablets als offline verwendbare Single-Page-Applikation,
- die zentrale boolesche Funktion soll im BEAST-Format exportierbar sein zur Import und Verwendung durch BEAST und
- den Dual-View-Modus, in dem, abhängig von Bildschirmgröße, links und rechts vom Menüband jeweils ein View angezeigt wird.

1.3 Mögliche Erweiterungen

Folgende Optionen zur Erweiterung von SANE werden beim Entwickeln bedacht:

- ein *Teacher-Modus*, welcher ermöglicht, dass bestimmte Views das zentrale Datenobjekt nicht ändern können. Anstelle dessen lassen sich Aussagen über die Korrektheit der vom Nutzer eingegebenen Daten treffen,
- ein View, der Struktur-Hasards innerhalb gefundener KV-Blöcke anzeigt und Möglichkeiten zur Vermeidung angibt. Die dafür angezeigte Schaltung könnte durch die Schnittstelle zu BEAST realisiert werden und
- einen g-Parameter als an- und ausschaltbare Option.

1.4 Abgrenzungskriterien

- Die SANE-Anwendung wird keine Startkonfiguration und damit keine den Lehrveranstaltungen ähnlichen Übungsaufgaben selbst generieren. Wird die Webseite nicht mit einer Startkonfiguration aufgerufen, werden alle Ausgangsfunktionen mit 0 initialisiert. Um den Nutzer mit SANE vertraut zu machen, werden einige, fest implementierte Beispiele angeboten.
- Da die Anwendung auch offline laufen muss, wird nach dem erstmaligen Laden der Seite kein Datenaustausch mit dem Server stattfinden.
- Es wird nur einen Nutzertyp geben, welcher als Standard für die Interaktion mit SANE definiert ist.
- Die Erzeugung von Startkonfigurationen geschieht nicht in der Anwendung selbst, sondern auf externe Weise, welche im Benutzerhandbuch beschrieben wird. Aus diesem Grund werden die Unterscheidung und das Management von Nutzern, sowie die Vergabe von Berechtigungen vermieden.

- Die Unterstützung von g-Parametern wird nicht implementiert, jedoch werden (GUI-) Schnittstellen und die Funktionen auf den Daten so gestaltet, dass diese später mit g-Parametern erweitert werden können.

2 Produkteinsatz

Die Software SANE soll in den Lehrveranstaltungen Technische Informatik, Rechnerorganisation und Schaltsysteme an der TU Ilmenau zur Vermittlung fachspezifischer Inhalte eingesetzt werden können. Die Studierenden sollen es weiterhin zum Selbststudium und Lösen der Seminar- und Praktikumsaufgaben verwenden.

2.1 Zielgruppen

Die Anwendung richtet sich besonders an Fächer, Studierende und Lehrende der TU Ilmenau, kann aber auf der Website auch von anderen Interessierten abgerufen werden, außerdem kann SANE von den Lehrenden zur Vermittlung des Stoffs genutzt werden.

2.2 Betriebsbedingungen

Für die einwandfreie Ausführung des Programms wird ein Computer benötigt, welcher einen Browser, vorzugsweise Chrome, ausführen kann. Der Computer benötigt einen Internetanschluss. Die Software soll jederzeit ausführbar sein. Es werden keine weiteren Programme wie Flashplayer oder Java benötigt.

3 Produktumgebung

SANE soll als Web-Applikation unabhängig vom Gerätetyp und sowohl auf PCs, als auch auf mobilen Geräten laufen und einfach bedienbar sein. Entwickelt wird es für Google Chrome, da er aktuell der am weitesten verbreitete Webbrowser ist und kostenlos für die meisten Plattformen angeboten wird, außerdem unterstützt er alle relevanten Standards. Die Hauptbibliothek der Anwendung, Polymer, wird von Google selbst entwickelt und wird wahrscheinlich auch längerfristig von Chrome unterstützt. Für andere Browser ist die volle Funktionalität nicht gewährleistet. Die fehlende Funktionalität kann aber über *polyfills* nachgeladen und emuliert werden. Diese Option ist vor allem für Firefox und Edge relevant.

4 Produktfunktionen

SANE soll die Lehrinhalte der o. g. Veranstaltungen mittels einer interaktiven Umsetzung der theoretischen Bestandteile und deren Zusammenhänge unterstützen.

Die einzelnen Funktionen werden in sogenannten Views bereitgestellt. Der Nutzer hat je nach Endgerät verschiedene Möglichkeiten, bestimmte Views auszuwählen und über diese Views mit der zentral gespeicherten booleschen Funktion in verschiedener Art und Weise zu interagieren.

Folgende Views werden dem Nutzer zur Verfügung gestellt:

1. Wertetabelle

Der Nutzer erhält die Möglichkeit eine Wertetabelle mit konfigurierbar vielen Eingangs- und Ausgangsvariablen zu erstellen. Der Wert einer Ausgangsvariable lässt sich durch einen Klick zwischen 1, 0 und * umschalten.

2. Karnaugh-Veitch-Diagramm

Die ausgewählte Ausgangsfunktion wird im Karnaugh-Veitch-Diagramm dargestellt. Es werden bis zu sechs Eingangsvariablen unterstützt. Auch hier lässt sich ein Bit mittels eines Klicks oder Tap zwischen 1, 0 und * umschalten. Im Diagramm lassen sich Blöcke bilden, um einen minimalen Ausdruck zu finden. Das Bilden von Blöcken ist dabei auch automatisch auf Grundlage des Quine-McCluskey-Algorithmus möglich.

3. Boolesches Mengendiagramm

Im booleschen Mengendiagramm lassen sich bis zu drei Ausgangsvariablen auswählen, die in dem Diagramm dargestellt werden sollen. Die Anzahl der dargestellten Elemente richtet sich dabei nach der Anzahl der Eingangsvariablen. Um die Übersichtlichkeit zu wahren, sollen maximal vier Eingangsvariablen, also 16 Elemente dargestellt werden. Der Nutzer kann durch Positionierung der Eingangsbelegungen die Funktion beeinflussen.

4. Boolesche Ausdrucksalgebra

Die zentrale boolesche Funktion wird in KDNF und KKNF sowie den minimalen KNF, DNF dargestellt. Alle minimalen Ausdrücke werden mit dem Quine-McCluskey-Algorithmus berechnet. Die Eingabe einer beliebigen booleschen Funktion führt zur Berechnung der minimalen Funktionen.

5. Quine-McCluskey-Algorithmus

Der QMC-Algorithmus wird mit allen Implikanten, Kürzungstabellen und Markierungen während der Minimierung angezeigt. Es werden weiterhin die Primimplikanten und die ausgewählten Implikanten dargestellt.

6. Funktions-Hasards

In einem KV-Diagramm werden die auftretenden Funktions-Hasards dargestellt.

7. Weitere Views

Es sind weiterhin folgende Views möglich:

a) Programmierbare Strukturen:

Dieser View dient zur Darstellung einer booleschen Funktion durch ROM, PLA und GAL.

b) Struktur-Hasards:

Es wird möglich sein, perspektivisch eine Schaltung aus BEAST mit SANE auf Struktur-Hasards zu analysieren.

In einem Settings-Fenster lassen sich globale Einstellungen treffen. Es lässt sich ein Zeichensatz für die Ausdrucksalgebra angeben, um das Arbeiten mit SANE zu vereinfachen. Der Anwendungszustand wird automatisch im *localStorage* gespeichert. Alternativ ist das exportieren auch manuell möglich.

Zur Verwendung in einer Lernumgebung wie Moodle soll es möglich sein, SANE mit einer Startkonfiguration zu versehen. Für alle Views ist es dann möglich das automatische Ausfüllen zu deaktivieren. Im Teacher-Modus soll nach dem Eintragen der Lösung eine automatische Überprüfung der Lösungen erfolgen. Die Studierenden sollen als Rückmeldung eine Popup-Nachricht mit der Korrektur erhalten. Für die Verwendung in anderen Systemen wird ein Log generiert.

5 Produktdaten

SANE speichert den Anwendungszustand automatisch im *localStorage* des Browsers. Die Funktion ist in allen gängigen Browsern verfügbar. Dadurch wird das Speichern auf allen Endgeräten mit einem aktuellen Browser möglich sein.

Es können heute circa fünf Megabyte in den meisten Browsern gespeichert werden. Zur Laufzeit der Anwendung kann davon ausgegangen werden, dass diese Grenze nicht überschritten wird, da nur einzelne Arrays mit der zentralen booleschen Funktion abgespeichert werden. Hierzu ist also nur die Speicherung einer überschaubaren Anzahl von Werten möglich.

Dank des *localStorage* ist es möglich, auf das Speichern von Cookies zu verzichten.

6 Produktleistungen

1. Weitergabe von Änderungen

Änderungen, die in einem View vorgenommen werden, sollen an andere Views weitergegeben werden.

2. Performanz

Die Anwendung soll performant sein, um auf Smartphones schnell ausgeführt werden zu können. Dies geschieht durch das Zwischenspeichern aufwendiger Berechnungen.

3. Sprachen

SANE soll sowohl in Deutsch als auch in Englisch verfügbar sein.

7 Vorgeschlagenes System

7.1 Benutzeroberfläche

7.1.1 Allgemeines zur graphischen Benutzeroberfläche

Das Kernelement der GUI ist die Hauptseite, über die alle anderen Funktionen der Anwendung erreicht werden sollen. Die Hauptseite zeigt die unterschiedlichen Views an, welche durch eine Navigation ausgewählt werden können. Zusätzlich gibt es noch eine Schaltfläche für die Einstellungen. Die Bestandteile der Hauptseite werden in Abbildung 7-1 dargestellt.

Welche Views beim Öffnen der Seite angezeigt werden, hängt von der Startkonfiguration oder der vorherigen Benutzung der Anwendung ab. Wurde die Anwendung bereits verwendet, so werden die zuletzt genutzten Views beim Öffnen angezeigt.

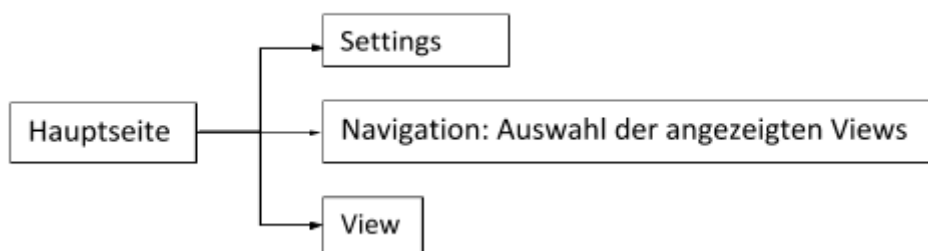


Abbildung 7-1: Bestandteile der Hauptseite

7.1.2 Details zur Benutzeroberfläche

Grundsätzlich wird das Layout der graphischen Benutzeroberfläche je nach Größe des Bildschirms des Endgerätes in mehreren Darstellungsweisen unterschieden, die im Folgenden genauer beschrieben werden.

Geräte mit kleiner Bildschirmgröße

Bei kleiner Bildschirmgröße (z. B. Smartphone) soll nur eine View (*Single-View*) angezeigt werden. Die Navigation lässt sich über eine Schaltfläche oder bei Touch-Geräten über einen Wisch vom linken Rand nach rechts öffnen. Die Schaltfläche für die Optionen befindet sich oben links.

Geräte mit größerer Bildschirmgröße

Bei größerer Bildschirmgröße (z. B. Desktop-PC) sollen zwei Views (*Dual-View*) gleichzeitig angezeigt werden. Die Navigation befindet sich permanent in der Mitte des Bildschirms. Die beiden Views sind links und rechts davon.

Anmerkungen zum Layout

Der Schwellenwert für Bildschirmgröße, der bestimmt, ab wann die Anwendung im *Single-* oder *Dual-View* dargestellt wird, soll erst im Verlauf der Implementierungsphase genau definiert werden.

Wird der Browser im *Dual-View* verkleinert und unter den Schwellenwert gebracht, so schaltet die Anwendung in der *Single-View* um.

7.2 funktionale Anforderungen

SANE_FA_01

View: Wertetabelle

Eine skalierbare Wertetabelle mit konfigurierbar vielen Eingangs- und Ausgangsvariablen sowie der Einbeziehung von h^* -Ausdrücken und umschaltbaren Ausgängen soll entworfen werden. Die Zeilenindizes sollen auf der linken Seite der Wertetabelle angezeigt werden.

sehr hohe Priorität

hoher Aufwand erwartet, ca. 52 Stunden

SANE_FA_02

View: Karnaugh-Veitch-Diagramm

Die zentrale boolesche Funktion soll in einem Karnaugh-Veitch-Diagramm (KV-Diagramm) dargestellt werden. Es sollen bis zu 6 Eingangsvariablen unterstützt werden. Ein Klick oder Tap auf ein Bit schaltet dieses zwischen 1, 0 und * um. Per Drag-and-drop, Shift-Klick und Long-Press Tap lassen sich Bits zur Vereinfachung des Ausdrucks

gruppieren. Weiterhin soll eine automatische Bildung der maximalen Blöcke in allen Varianten zur Ermittlung aller Minimalterme möglich sein. Grundlage für diese Berechnung stellt die Verwendung des Quine-McCluskey-Algorithmus dar.

hohe Priorität

hoher Aufwand erwartet, ca. 41 Stunden

SANE_FA_03

View: Boolesche Ausdrucksalgebra

Die zentrale boolesche Funktion wird in KDNF und KKNF sowie minimalen KNF und DNF in editierbarer Weise dargestellt. Alle minimalen Ausdrücke sollen ebenfalls durch QMC berechnet werden können. Eine Änderung des angezeigten Ausdrucks oder die Eingabe einer eigenen Gleichung passt die zentrale boolesche Funktion daran an.

hohe Priorität

mittlerer Aufwand erwartet, ca. 31 Stunden

SANE_FA_04

View: Quine-McCluskey-Algorithmus

Für die spezielle Ansicht des QMC-Algorithmus sollen alle Implikanten und Kürzungstabellen einschließlich Markierungen während der Minimierung angezeigt werden, sowie die Liste der Primimplikanten und der ausgewählten Implikanten für die endgültigen minimierten Terme.

hohe Priorität

mittlerer Aufwand erwartet, ca. 21 Stunden

SANE_FA_05

View: Funktionsindizes

Die Darstellung und Eingabe der zentralen booleschen Ausdrücke als Funktionsindizes soll ermöglicht werden. Die Ausdrücke werden dabei durch Hexadezimalzahlen bestimmt.

mittlere Priorität

geringer Aufwand erwartet, ca. 10 Stunden

SANE_FA_06

View: Boolesches Mengendiagramm

Die zentrale boolesche Funktion wird in einem Mengendiagramm dargestellt. Der Nutzer wählt bis zu 3 Ausgangsvariablen aus, die angezeigt werden. Die Anzahl der Elemente kann anhand der Anzahl der Eingangsvariablen bestimmt und verändert werden. Per Drag-and-drop lassen sich Werte zwischen den vorhandenen Mengen verschieben. Für die Übersichtlichkeit sollen maximal 4 Eingangsvariablen und 16 Elemente entsprechend der Belegung verwendet werden können.

mittlere Priorität

mittlerer Aufwand erwartet, ca. 31 Stunden

SANE_FA_07

View: Funktions-Hasards

Funktions-Hasards sollen erkannt und in einem KV-Diagramm dargestellt werden. Dabei wird das KV-Diagramm ohne die Funktionalitäten der KV-View eingeblendet und die Hasards darin angezeigt.

mittlere Priorität

mittlerer Aufwand erwartet, ca. 21 Stunden

SANE_FA_08

View: programmierbare Strukturen

Es sollen für die Darstellung ROM, PLA und GAL jeweils programmierbare und editierbare Strukturen als Ansichten der dahinterliegenden booleschen Funktion entwickelt werden. Durch Klick auf die einzelnen programmierbaren Verbindungen werden diese umgeschaltet. Die Anzahl der Eingangs- und Ausgangsvariablen ist in einem vordefinierten Rahmen einstellbar.

geringe Priorität

mittlerer Aufwand erwartet, ca. 21 Stunden

SANE_FA_09

View: Struktur-Hasards

Es sollen Struktur-Hasards innerhalb der gefundenen Blöcke im KV-Diagramm ermittelt werden, sowie deren Vermeidung. Für die Anzeige und Simulation der Schaltung aller Blöcke inklusive der redundanten Blöcke soll die in Anforderung SANE_FA_16 implementierte Schnittstelle zum BEAST verwendet werden.

sehr geringe Priorität

hoher Aufwand erwartet, ca. 62 Stunden

SANE_FA_10

Quine-McCluskey-Klasse

Der QMC-Algorithmus soll in ein wiederverwendbares Modul ausgelagert werden.

sehr hohe Priorität

hoher Aufwand erwartet, ca. 62 Stunden

SANE_FA_11

Settings

Das Settings-Menü dient zur Gruppierung der Einstellungen der einzelnen Views und zur Vermeidung von redundanten Einstellungen. Die in der Ausdrucksalgebra-View (nach SANE_FA_03) verwendeten Zeichen für Konjunktion, Disjunktion, Negation, Äquivalenz, Antivalenz und Implikation sollen konfigurierbar sein, mit Beschränkung auf 1-Zeichen-Ausdrücke. Das Erscheinungsbild der SANE-Webseite kann geändert werden.

hohe Priorität

mittlerer Aufwand erwartet, ca. 21 Stunden

SANE_FA_12

Progressive Webapplikation

Die Seite soll sich auf Smartphones und Tablet als offline-verwendbare Single-Page Applikation installieren lassen.

mittlere Priorität

geringer Aufwand erwartet, ca. 10 Stunden

SANE_FA_13

Export und Import des Anwendungszustandes

Die zentrale boolesche Funktion soll vom Nutzer exportiert und importiert werden können. Dazu wird die Funktion in einem geeigneten Format ausgegeben und dem Nutzer zum Speichern angeboten.

mittlere Priorität

geringer Aufwand erwartet, ca. 10 Stunden

SANE_FA_14

Übergabe einer Startkonfiguration

Mit dem Aufrufen der SANE-Webseite soll die Möglichkeit bestehen, eine Startkonfiguration vorzugeben, die den initialen Zustand der Anwendung beeinflusst. Dabei soll es möglich sein, bestimmte Views zuzulassen, zu beschränken und Funktionen der einzelnen Views auszuschalten. Die zentrale boolesche Funktion wird direkt auf die mitgegebenen Daten angepasst und die gewünschten Views angezeigt. Die Übergabe-Schnittstelle muss so definiert werden, dass sich die Webseite mit Startkonfiguration über einen Link öffnen oder mit einem iframe z. B. in Moodle einbinden lässt.

mittlere Priorität

hoher Aufwand erwartet, ca. 41 Stunden

SANE_FA_15

Teacher-Modus

Wenn der Teacher-Modus über die Startkonfiguration aktiviert wird, kann verhindert werden, dass bestimmte Views die zentrale boolesche Funktion ändern und stattdessen bei Änderungen überprüfen, ob der in der View vorliegende Ausdruck mit dem zentralen übereinstimmt. Der momentane Wahrheitswert der Übereinstimmung soll in einer extern abrufbaren Weise vorliegen.

sehr geringe Priorität

hoher Aufwand erwartet, ca. 41 Stunden

SANE_FA_16

Export im BEAST-Format

Die zentrale boolesche Funktion soll im BEAST-Format exportiert zur Verfügung stehen, sodass sie von BEAST importiert und verwendet werden kann.

geringe Priorität

mittlerer Aufwand erwartet, ca. 21 Stunden

SANE_FA_17

Support von g-Parametern

G-Parameter an sich nehmen sehr starken Einfluss auf alle Views, die zugrundeliegende globale Datenstruktur und die allgemeine Funktion des SANE-Projektes. Als Ausblick soll in der aktuellen Implementierung eine spätere Erweiterung um g-Parameter als umschaltbare Option berücksichtigt werden. Tabellen sollen eine Eingabe von mehreren möglichen Ausgängen und deren direkte Auswahl als aktiver Ausgang, sowie Karnaugh-Veitch-Diagramme die Eingabe eines logische g-Ausdruckes in eines der Felder etc. ermöglichen. Dieses Feature soll in diesem Softwareprojekt an sich noch nicht umgesetzt aber vorbereitet und bedacht werden.

sehr geringe Priorität

hoher Aufwand erwartet, ca. 62 Stunden

SANE_FA_18

Spracheinstellungen

Die Sprache der gesamten Anwendung soll zwischen Deutsch und Englisch umgeschaltet werden können. Dazu dient eine einfache Einstellmöglichkeit in den Settings (nach SANE_FA_11). Die Spracheinstellungen werden so implementiert, dass die angebotenen Sprachen erweiterbar sind.

sehr geringe Priorität

geringer Aufwand erwartet, ca. 10 Stunden

SANE_FA_19

Dual-View-Mode

Falls das Browser-Fenster groß genug ist, sollen zwei Views gleichzeitig nebeneinander angezeigt werden. Die Änderungen in einer View werden sofort auch in der entsprechenden Darstellung in der anderen View umgesetzt. Mittig zwischen den Views ist die Navigation dargestellt, die die Auswahl der Views sowie weitere Optionen enthält. Falls der Bildschirm nicht für den Dual-View-Mode geeignet ist, soll nur ein View angezeigt werden und das Menüband entsprechend "eingeklappt" werden.

mittlere Priorität

hoher Aufwand erwartet, ca. 52 Stunden

7.3 Nichtfunktionale Anforderungen

SANE_NFA_01

Polymer

Die zu entwickelnden Elemente werden mit Hilfe der JavaScript-Bibliothek Polymer implementiert. Die zu entwickelnden Komponenten werden als Web Components modularisiert und die endgültige Funktion über die Komposition bestehender und neu entwickelter Web Components erfüllt.

SANE_NFA_02

Material Design

Angelehnt an die wiederverwendeten Elemente aus der Polymer-Bibliothek rezipieren die neu entworfenen Elemente das Material Design.

SANE_NFA_03

Responsive Design

Alle Erweiterungen sollen ein responsives Design umsetzen. Das bedeutet, dass sie sich ebenso gut auf einem Desktop-Rechner wie auf einem Tablet oder Smartphone benutzen lassen. Das beinhaltet das Design ebenso wie die Reaktion auf Maus/Tastatur und Touchscreens.

SANE_NFA_04

Datenfluss

Der Datenfluss ist strikt unidirektional. Das bestehende Datenmodell wird genutzt und ggf. erweitert. Sämtliche Kommunikation findet von den Submodulen mittels Events statt. Änderungen des Anwendungszustandes werden mittels des Data-Bindings aus Polymer an die Submodule übergeben.

SANE_NFA_05

Separierung einer Event-Klasse

Zur Kommunikation von Änderungen des Anwendungszustandes (sog. Actions) senden Module Events, die an das oberste Modul übergeben werden, von dem sie verarbeitet werden. Bisher sind diese Actions statisch in die sie nutzenden Klassen verwoben. Sie sollen separiert werden, damit sie in neuen Klassen wiederverwendet werden können.

SANE_NFA_06

Modul für boolesche Operationen

Sämtliche Funktionen, Methoden und Operationen, die allgemeingültig und mehrfach benötigt werden, besonders für boolesche Operationen, sind in mindestens ein ES6-Modul auszulagern.

SANE_NFA_07

Nutzung von importHelpers

Um duplizierten Code der von TypeScript benötigten Funktionen zu vermeiden, soll TypeScript mit der Compiler-Option *importHelpers* ausgeführt und die Laufzeitbibliothek *tslib* einmalig geladen werden.

SANE_NFA_08

TypeScript

Sämtlicher JavaScript-Code ist in TypeScript zu entwickeln, damit Erweiterungen und Refactoring erleichtert werden. TypeScript bildet vor allem eine typisierte Obermenge von JavaScript. Alle Klassen, Methoden, Attribute und sonstige Variablen sollen typisiert sein.

SANE_NFA_09

moderne Plattformfunktionen

Der Quelltext soll in modernem JavaScript und mit aktuellem CSS entwickelt werden. Das entspricht ECMAScript 6 (auch ECMAScript 2015 oder ES6 genannt) und CSS 3. Mittels TypeScript können und sollen auch neuere JavaScript-Funktionen genutzt werden, da sie browser-kompatibel compiliert werden. Als Referenzplattform fungiert die aktuelle Version des Browsers Chrome. Bei der Entwicklung soll die Verwendung von JQuery explizit vermieden werden.

SANE_NFA_10

Performanz

Die Anwendung muss performant sein, um auch auf verbreiteten Smartphones schnell ausgeführt zu werden. Ergebnisse von aufwendigen Berechnungen sollen hierbei zwischengespeichert werden, um unnötige Rechenoperationen zu vermeiden.

7.4 Qualitätszielbestimmungen

	sehr wichtig	wichtig	weniger	unwichtig
Robustheit	X			
Zuverlässigkeit	X			
Korrektheit	X			
Benutzerfreundlichkeit		X		
Effizienz		X		
Portierbarkeit	X			
Kompatibilität			X	

7.5 globale Testfälle und -szenarien

Eine Aufstellung der globalen Testfälle und -szenarien befinden sich im Dokument „Testdrehbuch“.

7.6 Anwendungsfälle

Die Anwendung bietet dem Nutzer drei Grundfunktionen: Interaktion mit den einzelnen Views, Auswahl der angezeigten Views durch die Navigation und Zugriff auf gewisse Settings (SANE_FA_11).

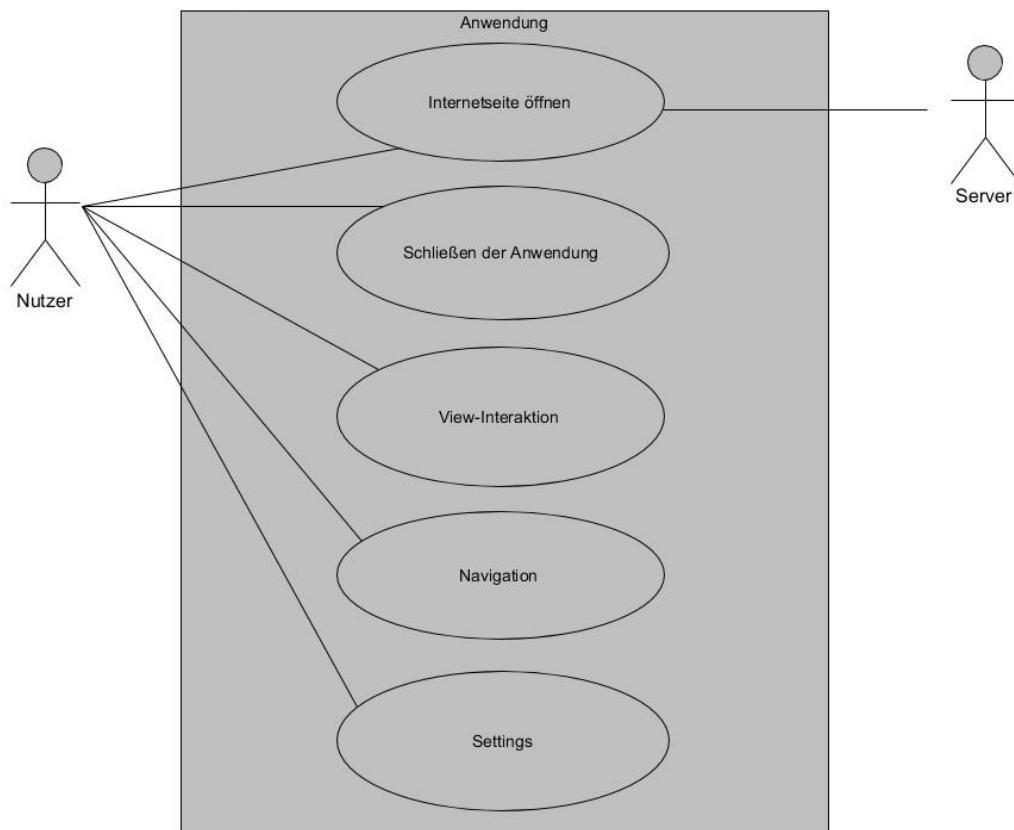


Abbildung 7-2: SANE

Die Folgenden Use-Case-Diagramme stellen Verfeinerungen der Settings und der einzelnen Views dar.

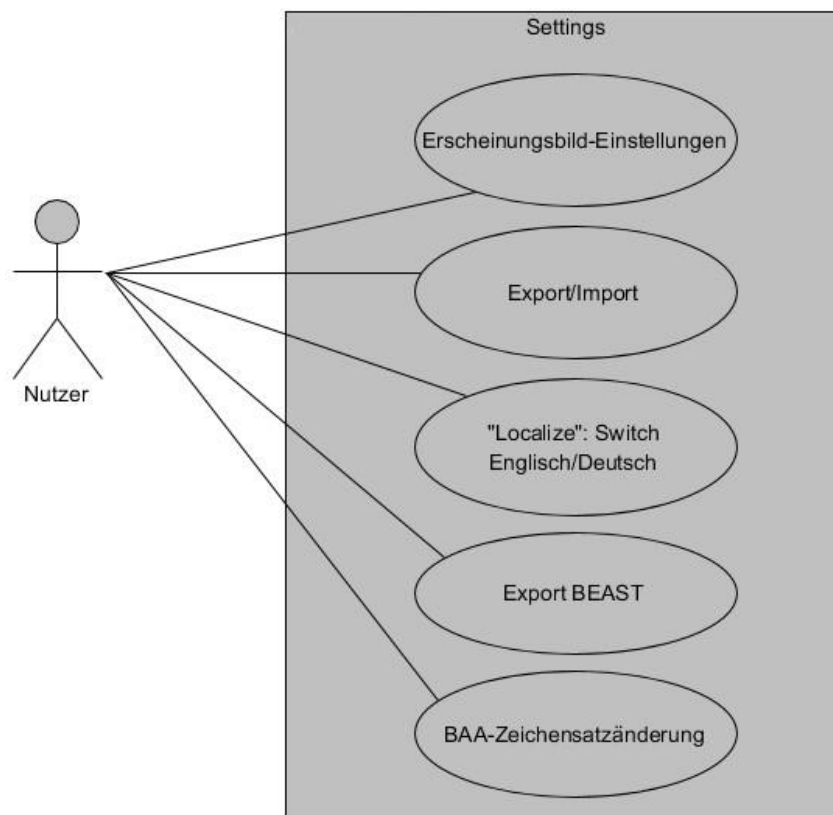


Abbildung 7-3: Die Settings-Funktion (SANE_FA_11)

Über die Schaltfläche *Settings* kann der Nutzer verschiedene, die Anwendung betreffende Einstellungen wählen bzw. Funktionen nutzen. Die Erscheinungsbild-Funktion soll es dem Nutzer möglich machen, zwischen Tag- und Nachtmodus zu wechseln. Die Import-/Exportfunktion (SANE_FA_13) dient zum Laden des Anwendungszustandes. Darüber hinaus soll auch ein Export auf die Anwendung BEAST in den Settings auswählbar sein. Klickt der Nutzer die Switch-Funktion, so soll die Anwendung zwischen den Sprachen Englisch und Deutsch umschalten. Die letzte Funktion ist die BAA-Zeichensatzänderung, die es dem Nutzer möglich macht, die verwendeten Zeichen für die Eingabe von Gleichungen selbst zu definieren.

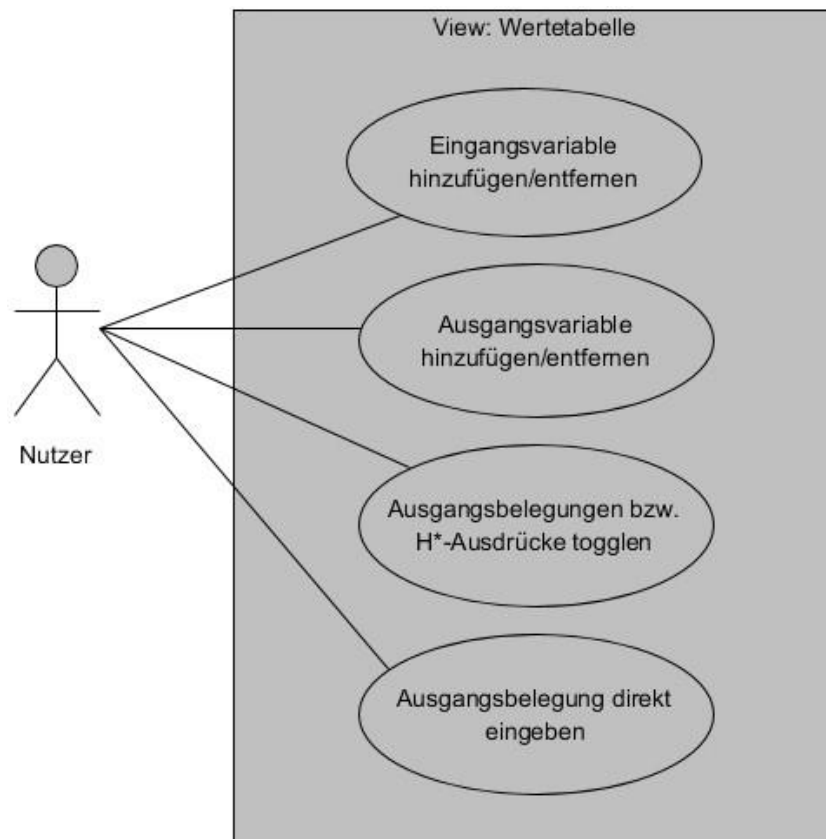


Abbildung 7-4: Wertetabelle (SANE_FA_01)

Der Benutzer erhält in der Wertetabelle-View (Abb. 7-4) die Möglichkeit, Ein- und Ausgangsvariablen hinzuzufügen oder zu entfernen. Außerdem kann er eine gewünschte Ausgangsbelegung direkt eingeben und durch Klicken mit der Maus bzw. Touch-Eingaben die einzelnen Bits der Ausgangsbelegung bzw. der h*-Ausdrücke zwischen 1 und 0 bzw. "leer" und * rotieren lassen.

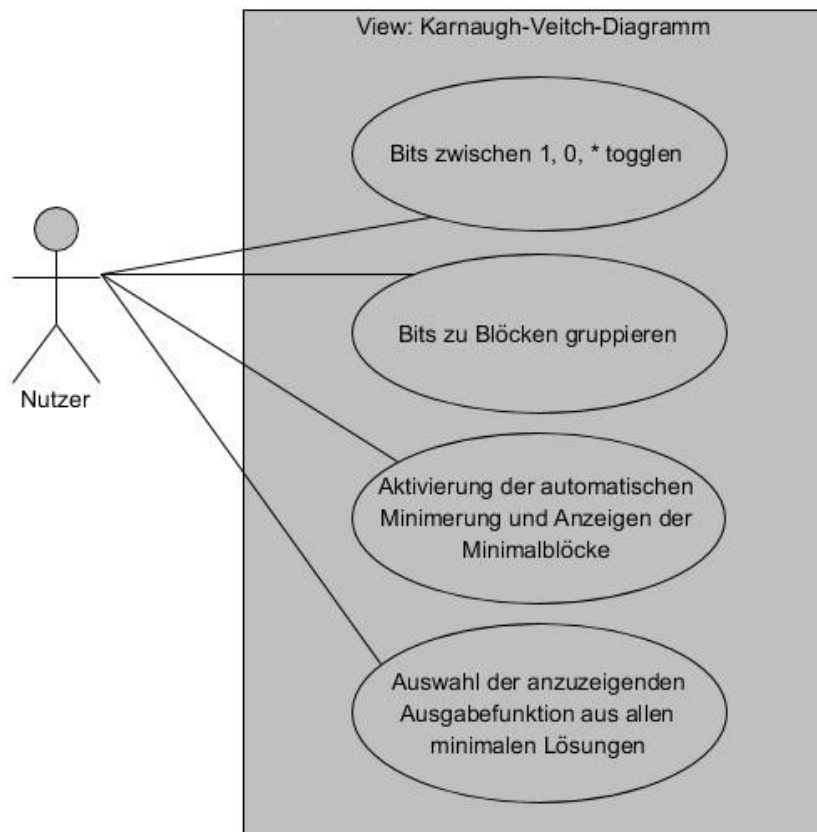


Abbildung 7-5: View: Karnaugh-Veitch-Diagramm (SANE_FA_02)

In der View zum Karnaugh-Veitch-Diagramm (Abb. 7-5) kann der Nutzer, wie auch in der View zur Wertetabelle, mit Hilfe der Maus oder durch Touch-Eingaben die einzelnen Einträge zu 0, 1 oder * verändern. Der Nutzer hat die Möglichkeit, die gewünschte Ausgabefunktion auszuwählen, damit diese dargestellt wird. Außerdem kann er die automatische Minimierung des Quine-McCluskey aktivieren oder auch manuell einzelne Bits zu Blöcken gruppieren, um den Ausdruck zu vereinfachen.

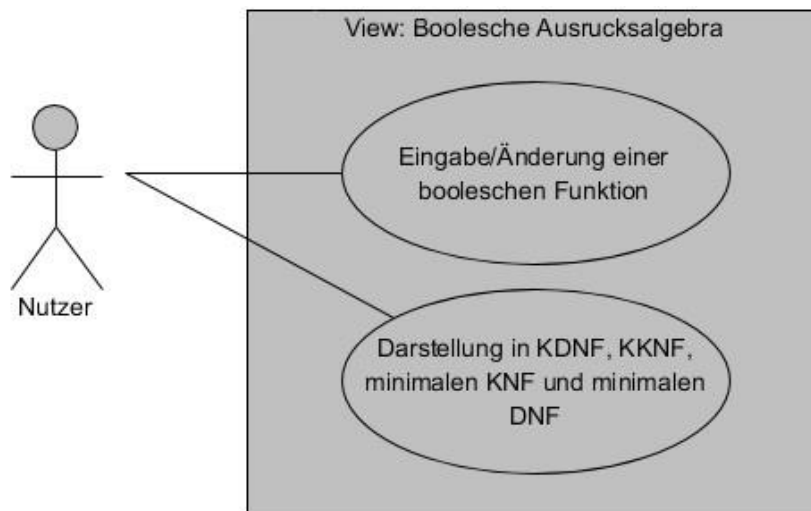


Abbildung 7-6: View: Boolesche Ausdrucksalgebra (SANE_FA_03)

In der View zur booleschen Ausdrucksalgebra (Abb. 7-6) wird die zentrale boolesche Funktion und der h^* -Ausdruck in KDNF, KKNF, minimaler KNF sowie DNF angezeigt. Änderungen des angezeigten Ausdrucks durch den Nutzer ändern auch die zentrale boolesche Funktion.

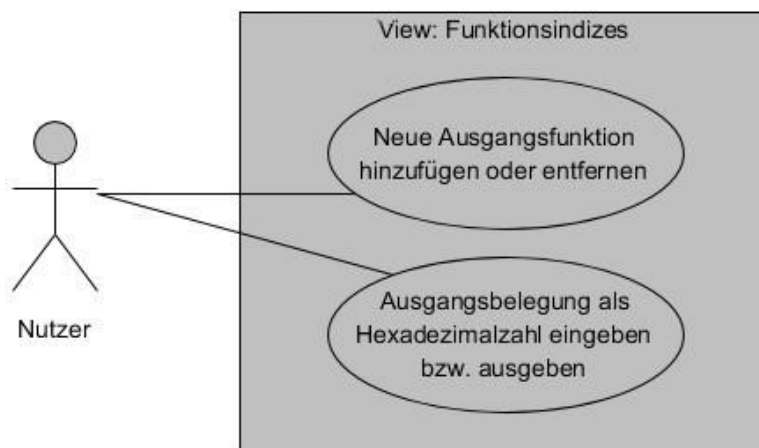


Abbildung 7-7: View: Funktionsindizes (SANE_FA_05)

Die View Funktionsindizes (Abb. 7-7) soll die Funktionsindizes aller vorhandenen Ausgangsfunktionen als Hexadezimalzahlen anzeigen. Der Nutzer hat dabei die Möglichkeit, neue Ausdrücke als Funktionsindizes einzugeben oder zu entfernen. Es gibt auch eine weitere Zeile für den h^* -Ausdruck.

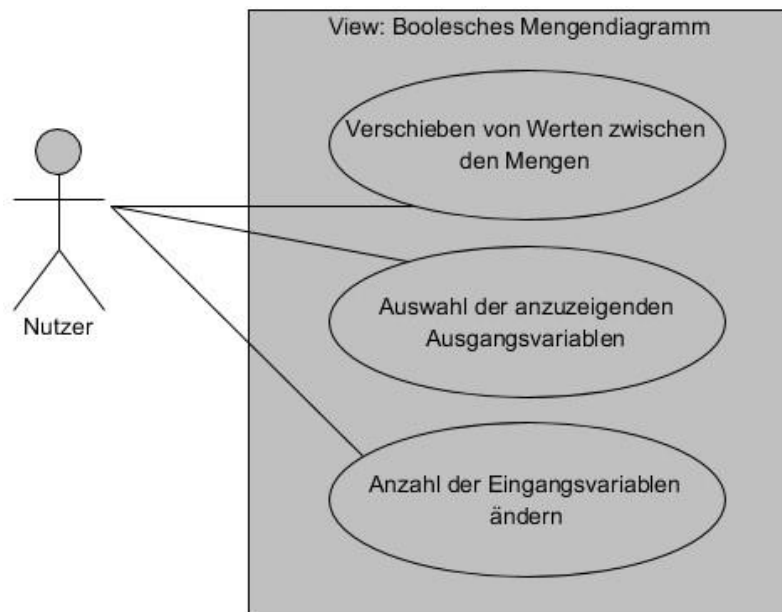


Abbildung 7-8: View: Boolesches Mengendiagramm (SANE_FA_06)

SANE soll es dem Nutzer auch möglich machen, die zentrale boolesche Funktion mit Hilfe von booleschen Mengendiagrammen nachvollziehen zu können. Dazu soll sich der Nutzer im gleichnamigen View (Abb. 7-8) die Mengen gewünschter Ausgangsvariablen anzeigen lassen können. Außerdem soll es möglich sein, Werte von einer Menge in eine andere zu verschieben und die Anzahl der Eingangsvariablen zu ändern.

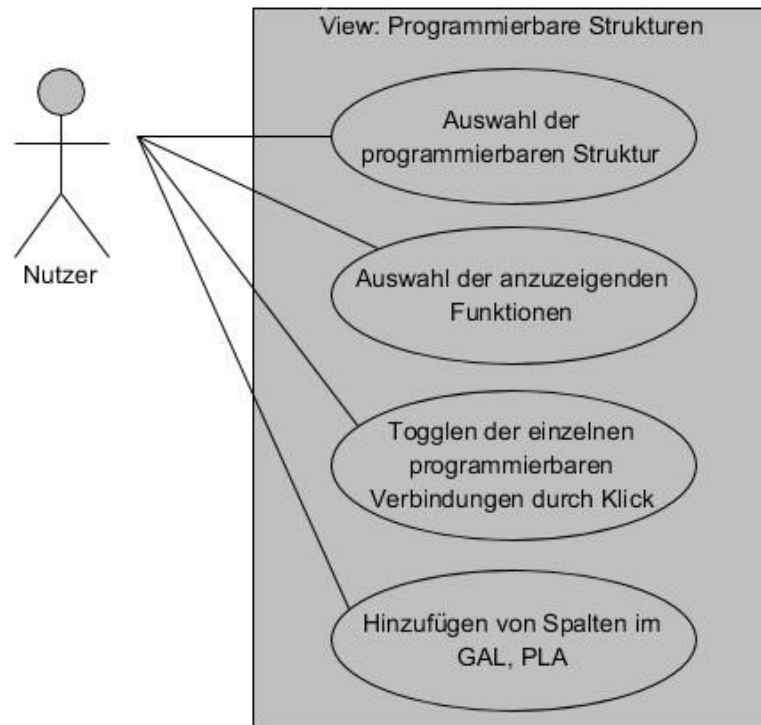


Abbildung 7-9: View: Programmierbare Strukturen (SANE_FA_08)

In der View zu den programmierbaren Strukturen (Abb. 7-9) kann der Nutzer sich die zentrale boolesche Funktion, realisiert mittels ROM, GAL oder PLA anzeigen lassen. Dabei soll es dem Nutzer möglich sein, durch ein Dropdown-Menü jeweils die gewünschte Struktur auszuwählen. Die einzelnen programmierbaren Verbindungen sollen durch Mausklicks oder Touch-Eingaben umgeschaltet werden. Der Nutzer kann außerdem die gewünschte Ausgabefunktion auswählen und manuell Spalten im PLA und GAL hinzufügen.

Anmerkung:

Die Views für Funktions- und Struktur-Hasards (SANE_FA_07 bzw. SANE_FA_09) sowie der View des Quine-McCluskey-Algorithmus (SANE_FA_04) wurden hier nicht aufgeführt, weil die einzige Nutzerinteraktion darin besteht, die gewünschte Ausgabefunktion auszuwählen.

8 Glossar

KDNF	Kanonische Disjunktive Normalform
DNF	Disjunktive Normalform
KKNF	Kanonische Konjunktive Normalform
KNF	Konjunktive Normalform
QMC	Quine-McCluskey-Algorithmus
ROM	Read Only Memory
PLA	Programmable Logic Array [Programmierbare logische Anordnung]
GAL	Generic Array Logic (äquivalent PAL)
PAL	Programmable Array Logic
BEAST	Block Diagram Editing and Simulating Tool
UP	Unified Process
DOM	Document Object Model