

Institut für Softwaretechnologie

Universität Stuttgart  
Universitätsstraße 38  
D-70569 Stuttgart

Bachelorarbeit Nr.257

# **Computer-aided Styleguides for Goal Structuring Notation**

Patrick Lux

<b>Studiengang:</b>	Softwaretechnik
<b>Prüfer/in:</b>	Prof. Dr. Lars Grunske
<b>Betreuer/in:</b>	Dr. Kai Höfig, Prof. Dr. Lars Grunske
<b>Beginn am:</b>	2015/6/1
<b>Beendet am:</b>	2015/10/15
<b>CR-Nummer:</b>	K.4.1, K.6.3



# Kurzfassung

Die vorliegende Bachelorarbeit befasst sich mit der Goal Structuring Notation (GSN) und deren Einsatz im sicherheitskritischen Bereich. Die GSN ist ein grafisches Stilmittel um Argumentationen wohlstrukturiert und anschaulich darstellen zu können.

Aufgrund steigender Komplexität in Soft- und Hardwaresystemen der heutigen Zeit wird der Nachweis der funktionalen Sicherheit immer undurchsichtiger und komplexer. In verschiedenen Branchen hat sich das Konzept der Safety-Cases durchgesetzt. Safety-Cases werden verwendet um Argumentationen über die Sicherheit eines Systems darzustellen. Diese Safety-Cases können auf verschiedene Arten modelliert werden, unter anderem mithilfe von Goal Structuring Notation Diagrammen.

In dieser Arbeit soll ein Ansatz zur Erstellung von Goal Structuring Notation Diagrammen entstehen und daraufhin implementiert werden. Durch Automatisierungen soll dem Anwender die Erstellung des Diagramms erleichtert werden. Zudem soll durch diese Automatisierungen auch die Qualität des Diagrammes gefordert werden. Dazu werden verschiedene Ansätze in Betracht gezogen.



# Abstract

In this bachelor's thesis the development and usage of Goal Structuring Notation is explored, especially in the context of displaying arguments of the functional safety of a given system. The GSN is a graphical Notation with the purpose of displaying arguments in a diagram structure.

The concept of safety-cases has become a widely used method for displaying how safety claims can be supported by given safety evidence for that specific system. Usually an argumentation will be provided to link the claims with the evidence.

Current Safety-Case displaying methods often use free text. Safety-Cases are a form of argumentation as already stated and thus can be displayed using the Goal Structuring Notation. A GSN diagram has quite a few advantages over simple free text considering the argument structure. Not only but especially relationships between statements can be displayed more visibly and clearly. Kelly and Weaver [2004]

For this purpose a tool to create and display GSN-Diagrams will be implemented. This tool will use the existing guidelines to help the user create a proper GSN-Diagram. In other words certain guidelines have to be met otherwise the Goal Structure is not considered proper or even legal. For this purpose the tool has to verify and validate the data input by the user to provide help and prevent structural and content related errors. In order to implement those features however they first need to be found and collected. It is therefore crucial for this thesis to find all guideline-related features that can be automated within the standards related to the topic. This should lead to an increased quality of the created GSN diagrams.

Additionally the question how much information is displayed throughout the GSN diagram and within the brackets itself shall be addressed.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Ziel der Arbeit . . . . .	2
1.3	Verwandte Arbeiten . . . . .	3
1.4	Struktur des Dokuments . . . . .	3
<b>2</b>	<b>Hintergrund und theoretisches Vorwissen</b>	<b>5</b>
2.1	Aufbau eines GSN-Diagramms . . . . .	6
2.2	Elemente der GSN . . . . .	7
2.3	Verknüpfungen . . . . .	11
2.4	Erstellung eines GSN-Diagramms . . . . .	13
2.5	Tim Kelly's Sechs-Schritt Methode . . . . .	13
2.6	Bottom-Up Methode . . . . .	16
2.7	Vermeidung häufig gemachter Fehler bei der Konstruktion eines GSN-Diagramms . . . . .	19
2.8	Erweiterte Elemente der Goal Structuring Notation . . . . .	21
2.9	Zusätzliche Informationen zur GSN-Diagramm Erstellung . . . . .	22
2.10	Evaluation eines GSN-Diagramms . . . . .	22
<b>3</b>	<b>Automatisierbare Eigenschaften</b>	<b>25</b>
3.1	Richtlinien und Indizien . . . . .	25
3.2	Automatisierungen . . . . .	28
<b>4</b>	<b>Die Implementierung</b>	<b>31</b>
4.1	Anforderungen . . . . .	31
4.2	Entwurf - Teil 1 - Speicherung der Daten . . . . .	32
4.3	Entwurf - Teil 2 - Das Stylesheet . . . . .	34
4.4	Entwurf - Teil 3 - Automatisierte XML Generierung . . . . .	35
4.5	Übersicht der Architektur . . . . .	35
4.6	Implementierung . . . . .	36
4.7	Aufbau der XML . . . . .	39
4.8	Der GSN-Generator . . . . .	40
4.9	verwendete Technologien . . . . .	40
4.10	Layout des GSN-Generators . . . . .	41

## Inhaltsverzeichnis

<b>5</b>	<b>Evaluation</b>	<b>43</b>
5.1	der Automatisierungen . . . . .	43
5.2	Tests . . . . .	44
5.3	Aufbau der Studie . . . . .	45
5.4	Hypothesen der Studie . . . . .	46
5.5	Durchführung der Studie . . . . .	47
5.6	Auswertung der Studie . . . . .	47
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>51</b>
6.1	Zusammenfassung . . . . .	51
6.2	Ausblick . . . . .	52
<b>A</b>	<b>Anhang A - Fragebogen der Fall- und Nutzerstudie</b>	<b>53</b>
<b>B</b>	<b>Anhang B - Zusammenfassung der Auswertung des Fragebogens</b>	<b>59</b>
<b>C</b>	<b>Abkürzungsverzeichnis</b>	<b>67</b>
	<b>Bibliography</b>	<b>73</b>

# Einleitung

In diesem Kapitel wird zunächst über die Motivation der Arbeit informiert, daraufhin gibt es eine Sektion zu den Zielen der Arbeit. Außerdem werden Arbeiten verwandter Themengebiete genannt.

## 1.1 Motivation

Unfälle innerhalb von Menschen gebauter Systeme gibt es schon seit tausenden von Jahren, wenn nicht schon seit Anbeginn der Menschheit. Innerhalb des letzten Jahrhunderts haben diese Systeme aber so große Dimensionen erlangt, dass Unfälle in Katastrophen ausarten können, Katastrophen die vielen Menschen das Leben kosten kann. Beispiele wie Tschernobyl und Fukushima zeigen, wie groß diese Dimensionen geworden sind. In Zeiten, bei denen ein Fehler einer einzigen Person das Leben tausender Menschen kosten kann, ist es wichtiger geworden denn je, die Sicherheit dieser Systeme zu bestimmen, zu analysieren und zu dokumentieren. Um Systeme generell sicherer zu machen wurden gewisse Standards und Richtlinien entworfen und entwickelt. Diese Standards können den Prozess der Erstellung definieren, Sicherheitsziele beinhalten oder anders bei der Erstellung beteiligt sein. Meist werden in den besagten Standards entstehende Dokumente oder Vorgehensweisen definiert, die entweder das Endprodukt oder den Erstellungsprozess dessen sicherer und überschaubarer macht.

In der Industrie wurden verschiedene Konzepte zur Reduktion der Sicherheitsrisiken sowie der Einhaltung von vorgegebenen Richtlinien und deren Dokumentation entwickelt. Einige dieser Konzepte haben sich mittlerweile etabliert, beispielsweise die sogenannten Safety-Cases. Der Zweck der Safety-Cases ist es, eine Verbindung des Beweismaterials und den Sicherheitszielen und Vorgaben durch ein Argument zu erreichen. In diesem Argument steht weshalb und warum ein Sicherheitsziel erreicht wurde, und wird mittels Beweismaterial belegt. Diese Verbindung zwischen Beweis und Sicherheitsziel ist selten trivial und erfordert einiges an Wissen und Können über das System.

Die Goal Structuring Notation ist eine Methode um Argumentationen grafisch, strukturiert und logisch konsistent darzustellen. Die Goal Structuring Notation hat verschiedene Anwendungsfelder, unter anderem kann sie dazu genutzt werden die bereits angesprochenen

## 1. Einleitung

"Safety-Cases" abzubilden. Kelly and Weaver [2004]

In der GSN gibt es verschiedene Elemente, deren Komposition eine Goal Structure ergeben. Eine Struktur die darauf ausgelegt ist, eine Behauptung mit Argumenten und Evidenzen zu belegen, sodass Zusammenhänge zwischen Evidenzen und Behauptungen gut sichtbar, transparent und logisch nachvollziehbar dargestellt werden können.

Die Komposition dieser Goal Structures ergibt ein Diagramm. In diesem Diagramm wird speziell der logische Fluss und die fundamentale Begründung der Argumentation hervorgehoben.

Diese Arbeit setzt an Publikationen Tim Kelly's an, der maßgeblich an der Entwicklung der Goal Structuring Notation beteiligt ist. Ausgehend von den bisher spezifizierten GSN-Notationen soll es im speziellen darum gehen, wie viele Informationen innerhalb der Blöcke der GSN dargestellt wird, und wie viel Information über die Hierarchietiefe hinweg abgebildet werden soll. Auch werden automatisierbare Eigenschaften gesucht. Das können unter anderem Stil Verbesserungen oder Richtlinien des GSN-Community Standards sein, die dem Benutzer bei der Erstellung eines GSN Diagrammes hilfreich sein können, oder eingehalten werden müssen.

## 1.2 Ziel der Arbeit

Das übergeordnete Ziel dieser Bachelorarbeit ist es, einen teil-automatisierten, modellbasierten Ansatz zur Modellierung wohlstrukturierter GSN-Diagramme zu entwickeln, der die im GSN-Community-Standard befindliche Reglementierung einhält und zudem weitere bisher undefinierte oder unzureichend beschriebene Richtlinien in Betracht zieht.

Ausgehend von den bisher spezifizierten GSN-Notationen soll es im speziellen darum gehen, wie viele Informationen innerhalb der Blöcke der GSN dargestellt wird, und wie viel Information über die Hierarchietiefe hinweg abgebildet werden soll. Auch werden automatisierbare Eigenschaften gesucht. Das können unter anderem Stil Verbesserungen oder Richtlinien des GSN-Community Standards sein, die dem Benutzer bei der Erstellung eines GSN Diagrammes hilfreich sein können, oder sogar eingehalten werden müssen.

Außerdem sollen automatisierbare Eigenschaften gefunden werden, die, falls für notwendig oder hilfreich erachtet, in dem Demonstrator implementiert werden. Für den Demonstrator soll ein Extensible Markup Language (XML)-basierter Ansatz zur Darstellung gewählt werden, sodass ein GSN-Diagramm innerhalb eines Browsers dargestellt werden kann.

### 1.3 Verwandte Arbeiten

Der Nachweis der funktionalen Sicherheit eines Systems ist mittlerweile ein sehr erforschtes Gebiet. Es gibt viele verschiedene Ansätze um die Sicherheit der immer komplexer werdenden Systeme zu bestimmen, zu analysieren und zu dokumentieren.

Zum einen gibt es immer genauer definierte Standards, wie beispielsweise der ISO 26262. Dieser Standard ist speziell darauf ausgelegt den Nachweis der funktionalen Sicherheit elektrischer Komponenten von Straßenfahrzeugen zu spezifizieren. Auf der anderen Seite wurden verschiedene Methoden zur Bestimmung von Sicherheitszielen und Vorgaben sowie der Dokumentation dieser entwickelt. Einerseits gibt es verschiedene Analysen vorhandener Sachverhalte, wie beispielsweise der Fault Tree Analysis (Fehler Baum Analyse, kurz FTA) oder auch der Failure Mode and Effects Analysis (kurz FMEA). Diese Analysen sind gute Methoden um Evidenzen eines Systems zu sammeln und zu dokumentieren. Happel et al. [2014]

Andererseits wurden auch verschiedene modellbasierte Notationen entwickelt um die gegebenen Dokumentationen sicherheitskritischer Komponenten besser darstellen zu können. Darunter beispielsweise die Architecture Analysis and Description Language (AADL) sowie die GSN. Kelly [1999b] Grunske and Han [2008]

Im engeren Themengebiet gibt es verschiedene Arbeiten zu Ziel basierten Notationen wie der Goal Structuring Notation. Beispielsweise einem Artikel über die Zukunft Ziel basierter Assurance-Cases. In diesem geht es vor allem darum, dass Ziel basierte Notationen durch Aufspaltung der Behauptungen sehr klar und akkurat Sachverhalte darstellen kann. Besonders betont werden die Relationen zwischen Argumenten, Beweis und Behauptungen. Bishop et al. [2004]

Im Sinne dieser Arbeit wird wie bereits erwähnt, hauptsächlich an den Publikationen Tim Kellys angeknüpft. Dieser ist maßgeblich an der Entwicklung der GSN beteiligt und hat einige Artikel darüber veröffentlicht. Viele dieser Arbeiten handeln von der Goal Structuring Notation. Einige wie der GSN Community Standard oder der Sechs-Schritt-Methode beschreiben den Aufbau und die Erstellung dieser GSN-Diagramme. Andere knüpfen an Safety- und Assurance-Cases an und betonen deren Relevanz innerhalb des Nachweises der funktionalen Sicherheit von Systemen. Kelly and Weaver [2004] Kelly [1999a] Kelly [2008] Weaver et al. [2003] Group et al. [2011] Kelly [1999b] Kelly [1998]

### 1.4 Struktur des Dokuments

Diese Bachelorarbeit behandelt das Thema Goal Structuring Notation, den damit verbundenen Richtlinien und Stil Vorgaben sowie der Semi- und Vollautomatisierung dieser in

## 1. Einleitung

einem Demonstrator.

Im bereits behandelten Kapitel, der Einleitung wurden bereits die Motivation und Ziele der Arbeit geklärt.

Im zweiten Kapitel, dem Hintergrund und theoretischen Vorwissen, wird es darum gehen, aus welchen Komponenten die Goal Structuring Notation besteht, wie diese aufgebaut ist und wofür sie im wesentlichen verwendet werden kann. Dazu werden die einzelnen Elemente der GSN, die innerhalb des GSN Community Standards und diverser anderer Veröffentlichungen beschrieben werden, analysiert und im Detail beschrieben. Außerdem wird es in diesem Kapitel um die Vorgehensweise zur Erstellung eines solchen Diagramms gehen. Im speziellen wird auf Tim Kellys "Sechs Schritt Methode" sowie eine Bottom-Up Methode zur Erstellung wohlstrukturierter GSN-Diagramme eingegangen.

Im dritten Kapitel der Arbeit geht es um die Automatisierbarkeit der im GSN Community Standard befindlichen Richtlinien und Stil Vorgaben. Im speziellen soll die Frage nach der Menge der Informationen innerhalb der Elemente sowie über die Hierarchietiefe hinweg geklärt werden. Außerdem sollen auch weitere automatisierbaren Eigenschaften gefunden werden, die mit der Implementierung des Werkzeugs zusammenhängen.

Im vierten Kapitel wird auf den Entwurf und die Implementierung des Werkzeugs eingegangen. Im speziellen wird auf die drei verwendeten Komponenten eingegangen, wie diese entworfen werden und welche Schritte der Arbeit diese bei der Erstellung und Darstellung eines GSN-Diagramms übernehmen.

Der fünfte Teil der Arbeit wird sich damit befassen, das entstandene Werkzeug zu evaluieren. Hierzu werden die Automatisierungen evaluiert, die Tests beschrieben, und eine Fall- und Nutzerstudie durchgeführt. Anhand der Resultate der Studie soll sich zeigen, ob das entwickelte Werkzeug benutzbar ist.

Der sechste und finale Teil der Arbeit beinhaltet die Zusammenfassung und den Ausblick.

# Hintergrund und theoretisches Vorwissen

Die heutigen Systeme sind üblicherweise eine Komposition kleinerer Teilsysteme die sich bereits in der Praxis bewährt haben, sodass der Kern der Argumentation, warum und weshalb ein System benutzbar sicher ist, zu großen Teilen auf der Architektur des Systems basiert. Durch den modularen Aufbau heutiger Systeme kann man die Argumentationen sowie deren Evidenzen der Teilsysteme mitverwenden um die Komposition dieser als sicher oder unzureichend sicher einzustufen. Kelly [2008]

Um die ausreichende Sicherheit eines Systems oder Teilsystems zu zeigen, kann ein Safety-Case verwendet werden. Ein Safety-Case soll ein klares, verständliches und zu verteidigendes Argument, dass ein System innerhalb eines gewissen Kontextes und Umgebung ausreichend sicher ist, präsentieren. Der Safety-Case besteht grundsätzlich aus drei Komponenten. Den Sicherheitsanforderungen des Systems, dem Beweismaterial, das mindestens einen Teil des Systems abdeckt und einem Argument, dass die Sicherheitsanforderungen mit dem Beweismaterial in Verbindung bringt.

Kelly [2008]

Evidenzen bzw. Beweisstücke sind unerlässlich beim Nachweis der funktionalen Sicherheit. Diese Beweisstücke können unterschiedlicher Natur sein. Zum einen gibt es vorgeschriebene Richtlinien die in Standards festgehalten werden und eingehalten werden müssen. In der Automobilindustrie wäre das beispielsweise die ISO 26262, einem Standard in dem die Vorgehensweise und der Nachweis der funktionalen Sicherheit elektrischer Komponenten von Straßenfahrzeugen spezifiziert ist. Zum anderen gibt es verschiedene Methoden um Indizien und Beweise der Sicherheit eines Systems zu sammeln und zu dokumentieren. Dazu gehören auch aber nicht ausschließlich Test Resultate, die FTA (Fault Tree Analysis) und die FMEA (Failure Mode and Effects Analysis) sowie deren Abwandlungen FMECA (Failure Mode Effect and Criticality Analysis) und FMEDA (Failure Modes, Effects and Diagnostic Coverage Analysis) Palin et al. [2011] Happel et al. [2014]

Argumente werden im allgemeinen dazu verwendet, Personen von einem Sachverhalt zu überzeugen. Im Fall eines Safety-Cases ist ein Argument die Schnittstelle zwischen Sicherheitsanforderungen und Beweismaterial. Die Aufgabe des Arguments ist es, Be-

## 2. Hintergrund und theoretisches Vorwissen

weismaterial und Sicherheitsanforderungen logisch konsistent miteinander zu verknüpfen, sodass der Leser den Sachverhalt versteht und nachvollziehen kann.

In der Praxis werden Safety-Cases oft als zusammenhängender Text beschrieben. Das kann aufgrund der Komplexität der Argumente dazu führen, dass schlecht strukturierte und somit schlecht nachvollziehbare Argumentationen entstehen. Um das zu vermeiden kann die Goal Structuring Notation als unterstützendes grafisches Hilfsmittel verwendet werden.

Das ursprünglich aus der nuklearen Branche stammende Modell der Goal Structuring Notation gilt mittlerweile als geeignetes und bewährtes Stilmittel um Argumentationen zielgerichtet und optisch ansprechend darstellen zu können.

Mithilfe der GSN können Safety-Cases leserlich und logisch konsistent bis hin zur Ebene der Evidenzen modelliert werden. Das ist besonders hilfreich, wenn die Kommunikation zwischen verschiedenen Fraktionen innerhalb des Herstellungsprozesses notwendig ist. Der ausschlaggebende Vorteil der Goal Structuring Notation ist, dass durch die grafische Darstellung fehlendes Beweismaterial und schlechte Begründungen sichtbar werden. Kelly [2008] Kelly and Weaver [2004]

### 2.1 Aufbau eines GSN-Diagramms

Ein Goal Structuring Notation Diagramm ist eine Komposition verschiedener Elemente, diese werden mit Verbindungen an einander gekoppelt, sodass eine zusammenhängende hierarchische Baumstruktur entsteht.

Das Schlüsselement in der GSN ist das Ziel (engl. Goal), das eine Behauptung oder Aussage enthält, die es im Verlauf des Diagramms zu belegen oder widerlegen gilt.

Dieses Ziel wird durch rekursive Schritte mittels Strategien in kleinere Teilziele aufgeteilt, bis hin zu einem Punkt an dem das vorliegenden Beweismaterial mit zuvor gemachten Behauptungen, direkt in Verbindung gebracht werden kann. Rückwirkend bedeutet das auch, dass der Nachweis aller bekannten Teilziele jeweils die Sicherheit der übergeordneten Ziele zeigt, und so, im Endeffekt, die Sicherheit eines kompletten Systems gezeigt werden kann. Anhand folgendem Beispiels soll der Sachverhalt und der Zusammenhang zwischen einzelnen Komponenten gezeigt werden.

## 2.2. Elemente der GSN

GSN COMMUNITY STANDARD VERSION 1

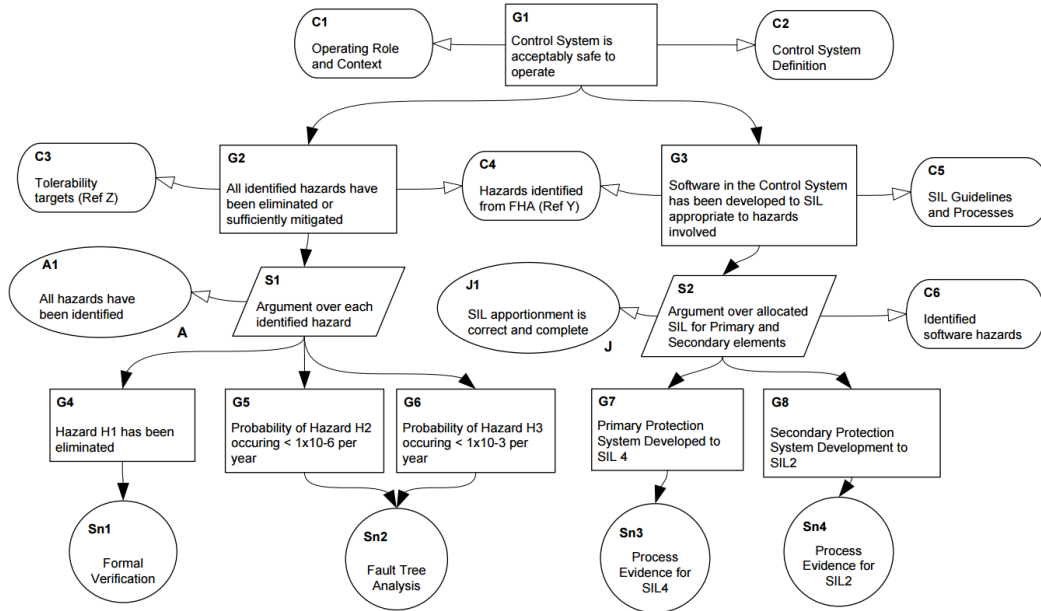


Abbildung 2.1. Beispiel aus dem GSN Community Standard Version 1

## 2.2 Elemente der GSN

Ein Element der Goal Structuring Notation ist grundsätzlich eine gezeichnete Struktur, dessen Inhalt auf Behauptung, Belegung oder kontextuelle Bedingung variiert. Diese Elemente werden in einem Goal Structuring Notation Diagramm verkettet, so dass, wie bereits erwähnt, eine Baumstruktur entsteht. Verkettete Elemente werden als Goal Structure bezeichnet.

Im folgenden sollen alle in der GSN verwendeten Elemente im beschrieben und erklärt werden.

### 2.2.1 Ziele

Ein Ziel der Goal Structuring Notation ist eine vom Autor getroffene Behauptung, die er belegen möchte. Diese Aussage gilt es über verschiedene Ebenen des Diagramms aufzuspalten und schließlich mit gegebenem Beweismaterial in Verbindung zu bringen.

## 2. Hintergrund und theoretisches Vorwissen

Beispiel eines Ziels wäre "System X ist ausreichend sicher".

In einem GSN-Diagramm existiert genau ein Ziel. Von diesem Ziel bzw. dieser Behauptung werden alle weiteren Behauptungen abgeleitet. Diese abgeleiteten Behauptungen werden innerhalb von Teilzielen dargestellt.

In begleitendem Umfang stehen meist zusätzliche Informationen innerhalb von Kontexten. Group et al. [2011] Spriggs [2012]

Innerhalb eines gezeichneten GSN-Diagrammes werden Ziele als Rechtecke dargestellt. Folgendes Beispiel soll den Aufbau eines Ziels veranschaulichen.

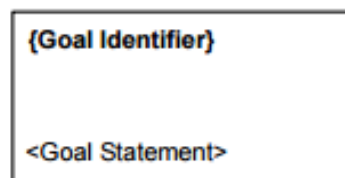


Abbildung 2.2. Beispiel eines Ziels aus dem GSN Community Standard Version 1

### 2.2.2 Teilziele

Ein Teilziel bzw. Subgoal ist eine von einem Ziel oder Teilziel höherer Ebene abgeleitete Behauptung. Die Behauptung in einem Teilziel deckt entweder durch die Aufspaltung einen Teilbereich der ursprünglichen Behauptung ab oder durch Umschreibung das gesamte Ziel bzw. Teilziel darüber gelegener Ebene. Group et al. [2011] Spriggs [2012]

Die Zeichnung und der Aufbau eines Teilzieles entsprechen dem eines Ziels.

### 2.2.3 Kontext

Ziele und Teilziele, bzw. deren Behauptungen, müssen immer in einem speziellen Kontext betrachtet werden. Der Kontext gibt unter anderem das Umfeld an, in dem sich ein System befindet. Der Kontext kann außerdem noch andere relevante Informationen erhalten. Darunter gehören unter anderem der Verweis auf verwendete Definitionen und die Erläuterung verwendeter Abkürzungen.

Zusammengefasst ist der Kontext ein Mittel, um relevante Zusatzinformationen über ein Ziel oder eine Strategie zu nennen. Meist wird ein Kontext dazu genutzt das Umfeld spezifisch zu Beschreiben und genauer zu deklarieren.

Group et al. [2011] Spriggs [2012]

Beispiel eines Kontextes wäre: "Die Abkürzung GSN bedeutet Goal Structuring Notation"

Folgende Abbildung soll den Aufbau und die Zeichnung eines Kontextes verdeutlichen.

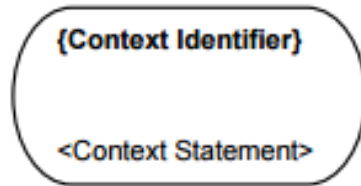


Abbildung 2.3. Beispiel eines Kontext aus dem GSN Community Standard Version 1

### 2.2.4 Strategien

Eine Strategie bzw. Strategy ist ein optionales Element. Die Strategie kann zwischen Ziel und Teilzielen bzw. Teilziel und Teilzielen darunter gelegener Ebenen gekoppelt werden, um die Vorgehensweise der Aufspaltung zu verdeutlichen. Sollte die Aufspaltung offensichtlich sein, kann auf eine Strategie verzichtet werden. Andernfalls sollte die aus Gründen der Verständlichkeit relevante Strategie beschrieben werden. Group et al. [2011] Spriggs [2012]

Beispiel einer Strategie wäre: "Es folgt ein Argument zu jeder identifizierten Gefahr"

Folgende Abbildung soll den Aufbau und die Zeichnung einer Strategie verdeutlichen.



Abbildung 2.4. Beispiel einer Strategie aus dem GSN Community Standard Version 1

### 2.2.5 Annahme

Eine Annahme bzw. Assumption wird in der Regel dann verwendet, wenn man nicht sicher ausschließen kann nicht alle Faktoren zu kennen oder externe Quellen zum Sicher-

## 2. Hintergrund und theoretisches Vorwissen

heitsnachweis verwendet und annimmt, das diese korrekt sind.

Beispiel einer Annahme wäre: "Alle Gefahren wurden identifiziert" Group et al. [2011] Spriggs [2012]

Folgende Abbildung soll den Aufbau und die Zeichnung einer Annahme verdeutlichen.

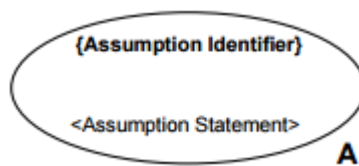


Abbildung 2.5. Beispiel einer Annahme aus dem GSN Community Standard Version 1

### 2.2.6 Rechtfertigungen

Eine Rechtfertigung bzw. Justification gilt meist der Unterstützung von Argumenten. Warum dieses Ziel oder diese Strategie aussagekräftig und verwendbar sind. Group et al. [2011] Spriggs [2012]

Beispiel einer Rechtfertigung wäre: " Die Safety Integrity Level (SIL) Aufteilung ist korrekt und vollständig"

Folgende Abbildung soll den Aufbau und die Zeichnung einer Rechtfertigung verdeutlichen.

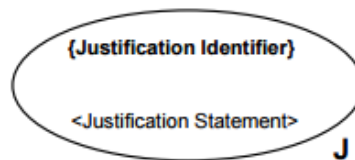


Abbildung 2.6. Beispiel eines Rechtfertigung aus dem GSN Community Standard Version 1

### 2.2.7 Lösungen

Eine Lösung bzw. Solution wird dazu verwendet, Beweise und Nachweise für Teilziele bzw. Subgoals zu referenzieren. Group et al. [2011] Spriggs [2012]

## 2.3. Verknüpfungen

Beispiel einer Lösung wäre: "Fault Tree Analysis". Zusätzlich kann innerhalb der Lösung der Ort, an dem sich das Dokument befindet referenziert werden.

Folgende Abbildung soll den Aufbau und die Zeichnung einer Lösung verdeutlichen.



Abbildung 2.7. Beispiel einer Lösung aus dem GSN Community Standard Version 1

### 2.2.8 Undeveloped Entity

Eine Undeveloped Entity oder auch unentwickelte Entität, im weiteren Verlauf als UDE abgekürzt ist ein stilistisches Mittel um noch nicht entwickelte Ziele und Strategien zu kennzeichnen. Die UDE ist laut dem GSN Community Standard speziell für den Fall gedacht, dass Ziele und Strategien absichtlich unentwickelt dargestellt werden können. Unter anderem auch für den Fall, dass Beweismaterial, was zum Zeitpunkt des Erstellens noch nicht vorhanden ist, ausgelassen werden kann. Dies kann nachträglich eingefügt werden, woraufhin die UDE entfernt wird.

Folgende Abbildung soll den Aufbau und die Zeichnung einer UDE verdeutlichen.

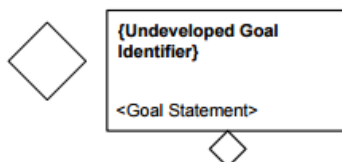


Abbildung 2.8. Beispiel einer UDE aus dem GSN Community Standard Version 1

## 2.3 Verknüpfungen

Es gibt zwei Arten von Verknüpfungen eine Kontext-Von Beziehung sowie eine Unterstützt-Von. Elemente die verknüpft werden sollen haben genau ein Vaterelement. Es gibt jedoch ein Ausnahmefall, die Lösung. Mehrere Teilziele können eine Unterstützt-von Verknüpfung

## 2. Hintergrund und theoretisches Vorwissen

zu einer Lösung besitzen, da eine Lösung möglicherweise Evidenz für mehrere Argumente sein kann. Group et al. [2011]

### 2.3.1 In-Kontext-Von Verknüpfung

Die In-Kontext-Von-Beziehung wird für alle kontextuellen Verknüpfungen genutzt. Unter kontextuelle Verknüpfungen sind alle Verknüpfungen gemeint, die relevante Zusatzinformationen enthalten, darunter der Kontext, die Rechtfertigung sowie die Annahme. Group et al. [2011]



**Abbildung 2.9.** Beispiel einer Kontext-Von Verknüpfung aus dem GSN Community Standard Version 1

#### legale Anwendungsfälle der In-Kontext-Von Verknüpfung

Im folgenden wird eine Liste mit allen legalen Anwendungsfällen von In-Kontext-Von Verknüpfungen genannt.

- ▷ Ziel zu Kontext
- ▷ Ziel zu Rechtfertigung
- ▷ Ziel zu Annahme
- ▷ Teilziel zu Kontext
- ▷ Teilziel zu Rechtfertigung
- ▷ Teilziel zu Annahme
- ▷ Strategie zu Kontext
- ▷ Strategie zu Rechtfertigung
- ▷ Strategie zu Annahme

### 2.3.2 Unterstützt-Von Verknüpfung

Unterstützt-Von Verknüpfungen sind Referenz bzw. Beweis Beziehungen zwischen Zielen, Teilzielen Strategien und Lösungen. Die Allgemeine Zeichenweise entsprechend dem GSN-Community Standard ist ein ausgefüllter Pfeil, wie in folgender Abbildung beschrieben.

## 2.4. Erstellung eines GSN-Diagramms



**Abbildung 2.10.** Beispiel einer Unterstützt-Von Verknüpfung aus dem GSN Community Standard Version 1

### legale Anwendungsfälle der Unterstützt-Von Verknüpfung

Im folgenden wird eine Liste mit allen legalen Anwendungsfällen von Unterstützt-Von Verknüpfungen genannt.

- ▷ Ziel zu Ziel
- ▷ Ziel zu Strategie
- ▷ Strategie zu Ziel
- ▷ Ziel zu Lösung

## 2.4 Erstellung eines GSN-Diagramms

Die Erstellung eines GSN-Diagrammes wird in den folgenden 2 Sektionen anhand von zwei im GSN Community Standard so wie diverser anderer Artikel Tim Kellys beschriebenen Methoden detailliert beschrieben.

## 2.5 Tim Kelly's Sechs-Schritt Methode

Die Sechs-Schritt Methode dient zur Entwicklung eines Goal Structuring Notation Diagramms. Diese "Top-Down" Methode wurde von Tim Kelly 1999 entwickelt und in einem Techreport unter dem Namen A Six-Step Method for Developing Arguments in the Goal Structuring Notation veröffentlicht Kelly [1999a].

Der generelle Zweck der GSN ist, wie bereits erwähnt, die Aufspaltung der Ziele in Teilziele, bis zu einer Ebene bei der mit Beweismitteln Argumentiert werden kann warum besagte Teilziele belegt wurden. Folgender Algorithmus bezieht sich auf die Vorgehensweise und Erstellung eines konformen GSN-Diagramms. Kelly [1999a] Group et al. [2011]

Die Aussagen in diesem Kapitel sind aus Tim Kellys Techreport sowie aus dem GSN Community Standard entnommen und als solche gekennzeichnet.

## 2. Hintergrund und theoretisches Vorwissen

1. Erstellung des Ziels.
2. Deklaration des Kontextes des Ziels.
3. Erstellung der Strategie.
4. Deklaration des Kontextes der Strategie.
5. Ausarbeitung der Strategie und Erstellung von Teilzielen.
6. Erstellung einer Lösung.

Im folgenden sollen die sechs Schritte im Detail erklärt werden.

### 2.5.1 Schritt 1 - Erstellung des Ziels

Zu Beginn steht der Autor vor der Herausforderung ein Ziel höchster Ebene zu erstellen, ein sogenanntes "Top Goal" oder auch "Top-Level Goal". Dieses Ziel enthält die grundsätzliche Aussage des Autors, die es im Verlauf des Diagrammes bzw. der Argumentation zu Zeigen gilt. Ein mögliches Top-Goal wäre beispielsweise "Die Benutzung des Systems X ist ausreichend sicher" Kelly [1999a].

An dieser Stelle ist die höchste Priorität, dem zukünftigen Leser die Aussage des Autors nahe zu bringen, und bewusst zu machen auf was der Autor hinaus möchte. Die Aussage muss den Leser überzeugen können Kelly [1999a].

Entstehende Teilziele werden von Zielen auf höheren Ebenen abgeleitet. Hierzu kann eine Strategie nötig sein, dazu mehr in folgenden Abschnitten.

Mit diesem Wissen wird nun das Ziel erstellt.

### 2.5.2 Schritt 2 - Deklaration des Kontextes des Ziels

Unter der Deklaration des Kontextes des Ziels ist im besonderen die Umgebung und der Kontext sowie sämtlichen relevanten zusätzlichen Informationen eines Ziels gemeint.

Ein Ziel soll im Verlauf der Argumentation vom Leser als erfüllt oder nicht erfüllt, einfacher als wahr, oder falsch erkannt werden können. Um sicherzustellen, dass der Leser ein klares Bild von der Problemstellung sowie dessen Lösung hat, müssen alle relevanten Informationen vom Autor in Kontexten verarbeitet werden. Group et al. [2011]

### **2.5.3 Schritt 3 - Erstellung der Strategie**

Wurde ein Ziel definiert und mittels Kontext die Rahmenbedingungen bestimmt folgt der nächste Schritt, die Aufspaltung des Ziels in kleinere, leichter zu belegende, Teilziele. Ziele aufzuspalten ist oft keine triviale Angelegenheit, deshalb kann es von Vorteil sein, die Vorgehensweise des Autors zu dokumentieren und Sie dem Leser innerhalb einer Strategie zugänglich zu machen. Group et al. [2011]

Es gibt verschiedene Herangehensweisen bei der Aufspaltung, zum einen das Prinzip "Divide and Conquer". Ziele werden in kleinere Teilziele aufgeteilt, wobei diese zusammen genommen die selbe Aussage treffen wie die des ursprünglichen Ziels aber dennoch einfacher mit aussagekräftigen Evidenzen belegbar sind. Eine andere Herangehensweise ist es, Ziele zu Umschreiben, sodass die Aussage zwar prinzipiell die Gleiche ist, doch anders Argumentiert werden kann.

### **2.5.4 Schritt 4 - Deklaration des Kontextes der Strategie**

Wie schon im Schritt 2 beschrieben, müssen Systeme immer in einem gewissen Kontext und Umfeld betrachtet werden, das Selbe gilt auch für Strategien. Auch diese sind nur in einem speziellen Kontext und Umfeld valide, und da der Autor an den Leser gebunden ist, müssen auch im Fall der Strategie alle zum Verständnis und Klarstellung notwendigen Informationen angegeben werden. Group et al. [2011]

### **2.5.5 Schritt 5 - Ausarbeitung der Strategie und Erstellung von Teilzielen**

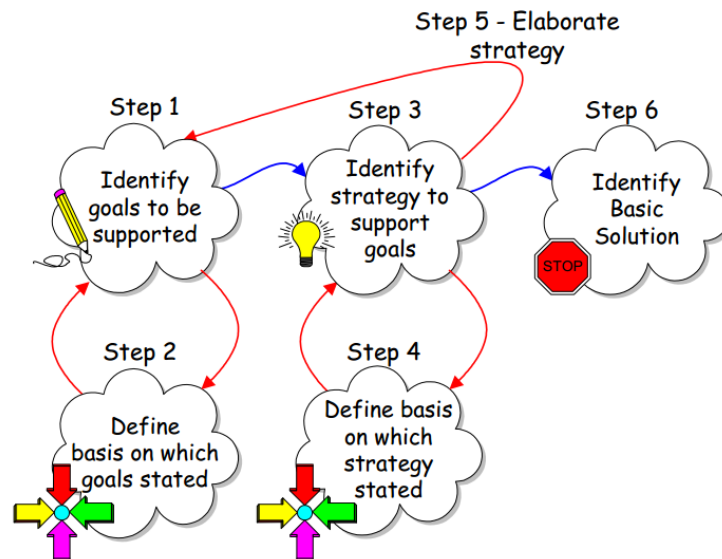
Schritt 5 ist der rekursive Schritt des Algorithmus. Nachdem eine Strategie entwickelt wurde um Ziele aufzuspalten kommt der Autor an den Punkt, an dem er entscheiden muss, ob diese Ziele nun mit einer Lösung bzw. Beweismaterial in Verbindung gebracht werden können, oder ob dazu weitere Aufspaltungen notwendig ist. Sollten die Ziele deutlich genug mit dem Beweismaterial in Verbindung gebracht werden können, geht der Autor in Schritt 6 über. Sollte eine weitere Rekursion benötigt werden beginnt der Autor wieder in Schritt 1. Group et al. [2011]

### **2.5.6 Schritt 6 - Erstellung einer Lösung**

Ist ein Ziel auf einer Ebene angelangt, auf der es direkt mit einem Beweisstück in Verbindung gebracht werden kann, so kann der Autor einen Lösungsblock erstellen,

## 2. Hintergrund und theoretisches Vorwissen

in dem er auf das Beweisstück referenziert. In dem Fall ist das verknüpfte Ziel im GSN-Diagramm abgeschlossen. Sind Alle Ziele im GSN-Diagramm über eine oder mehrere Ebenen hinweg mit Evidenzen verknüpft so ist das Diagramm abgeschlossen. Group et al. [2011]



**Abbildung 2.11.** Eine Skizze der Vorgehensweise in der Sechs-Schritt Methode entnommen aus Kelly [1999a].

## 2.6 Bottom-Up Methode

Es gibt weitere Varianten der Sechs-Schritt Methode zur Erstellung korrekter GSN-Diagramme eingesetzt werden können, unter anderem eine rekursive "Bottom-Up" Methode. Anders als bei der Sechs-Schritt Methode, wird zu erst das Beweismaterial gesammelt und begutachtet. Daraufhin werden Lösungen erstellt, in denen die Beweise referenziert werden. Darauf folgend werden Ziele niedrigster Ebene erstellt, die direkt mit dem Beweismaterial in Verbindung stehen. In einem rekursiven Schritt folgt nun die Erstellung einer Strategie, wie ein oder mehrere Ziele in ein Ziel darüber gelegener Ebene zusammengefasst oder umschrieben werden kann. Group et al. [2011]

Im Detail:

## 2.6. Bottom-Up Methode

1. Evidenzen identifizieren und als Lösungen präsentieren.
2. Identifizierung der Aussagen der Evidenzen, Verbindung mit Zielen.
3. Erstellung höher gelegener Teilziele.
4. Entwurf einer Strategie.
5. Kontext Erstellung.
6. Überprüfung der Vollständigkeit.
7. Injektion des entstandenen GSN-Zweigs in ein bereits existierendes GSN-Diagramm

### 2.6.1 Schritt 1 - Evidenzen identifizieren und als Lösungen präsentieren

Um die Lösungsebene nach der Bottom-Up Methode zu erstellen, muss der Autor die ihm vorliegenden Evidenzen in Lösungen referenzieren. Group et al. [2011]

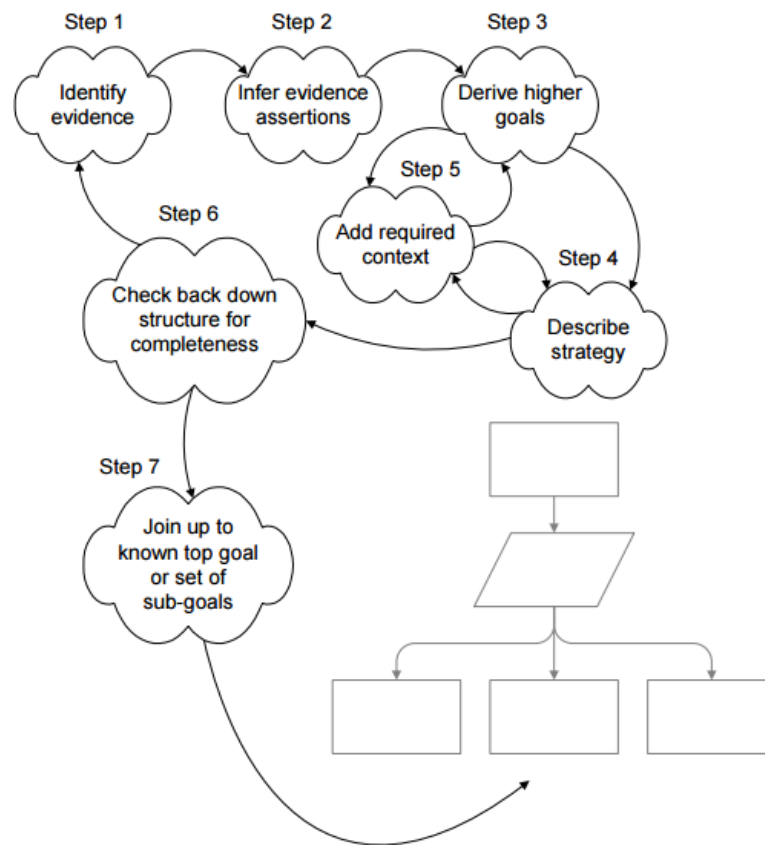
### 2.6.2 Schritt 2 - Identifizierung der Aussagen der Evidenzen, Verbindung mit Zielen

Der zweite Schritt dient der Identifizierung der Aussagen der Evidenzen. Die Evidenzen sind meist Nachweise einer Behauptung von einer oder mehreren Komponenten, des betrachteten Systems. Diese Behauptungen müssen als Teilziele ausformuliert und daraufhin mit den Lösungen verbunden werden. Inhaltlich sowie innerhalb der Darstellung. Group et al. [2011]

### 2.6.3 Schritt 3 - Erstellung der Teilziele höherer Ebenen

Wie der Name des Schrittes von sich gibt, geht es darum wie die bereits entworfenen Teilziele auf höhere Ebenen abstrahiert werden können. Wie Teilziele kombiniert werden können, um Ziele höherer Ebenen zu entwerfen wird oft mit einer "Trial and Error" Methode herausgefunden. Das Ziel ist es, einen logisch konsistenter Übergang innerhalb der Abstraktionsebenen zu finden. Eine Abstraktion zu hohen Grades, kann Konsequenzen auf den logischen Fluss innerhalb des Diagrammes haben und sich somit negativ auf das Diagramm auswirken. Group et al. [2011]

## 2. Hintergrund und theoretisches Vorwissen



**Abbildung 2.12.** Eine Skizze der Vorgehensweise in der Bottem-Up Methode entnommen aus Group et al. [2011].

### 2.6.4 Schritt 4 - Entwurf einer Strategie

Der Entwurf einer Strategie zur Verbesserung der Leserlichkeit und konsistenteren Darstellung des logischen Flusses führen. Eine Strategie ist wie bereits beschrieben optional. Der Autor sollte aber in jedem Fall sicher gehen, dass der Leser des Diagramms die Aufspaltung versteht. Im speziellen müssen Zweideutigkeiten vermieden werden. Group et al. [2011]

Wichtig ist dabei zu beachten, dass die Strategie nicht beinhaltet wie Teilziele abstrahiert wurden, sondern wie das Ziel höherer Ebene in die Teilziele niedrigerer Ebenen aufgeteilt werden kann. Group et al. [2011]

## 2.7. Vermeidung häufig gemachter Fehler bei der Konstruktion eines GSN-Diagramms

### 2.6.5 Schritt 5 - Kontext Erstellung

Die Erstellung der Kontexte mit jeglichen relevanten Zusatzinformationen, darunter aber nicht ausschließlich Definitionen verwendeter Abkürzungen, Annahmen und Referenzen. Group et al. [2011]

### 2.6.6 Schritt 6 - Überprüfung der Vollständigkeit

Die Überprüfung auf Vollständigkeit ist der kritische Schritt der "Bottom-Up" Methode. Durch die Überprüfung werden ausgelassene oder schlecht spezifizierte Elemente erkannt und können geändert werden. Im besonderen soll darauf geachtet werden, dass Teilziele ihre Eltern auch vollständig abdecken. Falls dem nicht so ist können ohne größere Probleme im Nachhinein neue Teilziele erstellt werden, sofern das Beweismaterial auch den Nachweis dieser Aussagen liefert. Group et al. [2011]

### 2.6.7 Schritt 7 - Injektion

Die Bottom-Up Methode wird üblicherweise nur für Teilkomponenten eines GSN-Diagrammes verwendet, da die "Top-Down" Methode generell unkomplizierter ist. Die Injektion des entstandenen GSN-Zweigs in ein bereits existierendes GSN-Diagramm erfolgt über eine Schnittstelle, bei der der Ersteller sichergehen muss, dass die bereits vorhandenen Komponenten des GSN-Diagramms ineinander greifen und somit der logische Fluss erhalten bleibt. Group et al. [2011]

## 2.7 Vermeidung häufig gemachter Fehler bei der Konstruktion eines GSN-Diagramms

### 2.7.1 Zweideutigkeit

Die Zweideutigkeit eines Wortes oder Satzes sollte in den meisten Fällen verhindert werden. Semantische Zweideutigkeit führt nicht selten zu einem falschen Verständnis vorliegender Tatsachen und das wiederum führt dazu, dass der Leser des Diagramms der Argumentation nicht folgen kann. Kelly [1999a] Group et al. [2011]

## 2. Hintergrund und theoretisches Vorwissen

### 2.7.2 Vagheit

Oft werden Wörter benutzt, die essentiell bedeutungslos sind. Diese Wörter dienen dazu Sätze schöner zu gestalten oder zu umranden. Wörter wie beispielsweise "einfach", "angemessen", "oft" und "effektiv" sagen relativ wenig über den vorhandenen Sachverhalt aus, außer sie werden innerhalb eines Kontextes quantitativ oder qualitativ definiert. Kelly [1999a] Group et al. [2011]

### 2.7.3 Zu starke Vereinfachung

Zu Starke Vereinfachung kann zu Vagheit führen und sollte deshalb vermieden werden. Andernfalls kann ein zu einfach beschriebenes Ziel auch dazu führen, dass die Argumentation nicht zu Ende geführt werden kann, weil unbeantwortbare Lücken im Diagramm entstehen. Kelly [1999a] Group et al. [2011]

### 2.7.4 Voraus greifen

als Ziel bzw. Top-Level Goal muss das fundamentale Ziel der Argumentation genannt werden. Andernfalls kann es dazu kommen, dass ein Teilbereich des eigentlichen Ziels nicht durch die Argumentation abgedeckt wird. Das kann zu besonders schlechten Umständen führen sofern es dem Autor nicht auffällt. Bleibt es dem Autor verborgen so kann es dazu kommen, dass dieser annimmt alle Fehlerquellen dokumentiert zu haben, wobei in der Realität ein Teilbereich des Ziels durch voraus greifen nicht in Betracht gezogen wurde. Das führt zu unentdeckten Lücken in der Argumentation, sodass die Sicherheit des Systems nicht mehr gewährleistet werden kann. Group et al. [2011]

### 2.7.5 Sprünge

Sprünge innerhalb der Argumentation können beispielsweise durch das Auslassen einer Strategie entstehen. In einem solchen Fall kann es dazu kommen, dass der Autor die falsche Annahme macht, dass der Leser im folgen kann, wobei dieser in Wirklichkeit längst den Faden verloren hat. Group et al. [2011]

### 2.7.6 weitere Fehlerquellen

Nicht zuletzt können falsch genutzte Kontexte, Strategien und andere Elemente dazu führen, die Struktur der GSN negativ beeinträchtigen. Der Autor sollte sich grundsätzlich an die Definitionen der Elemente halten um Fehler dieser Art zu vermeiden. Group et al.

[2011]

## 2.8 Erweiterte Elemente der Goal Structuring Notation

Für verschiedene Anwendungen der GSN in der Industrie wurden Erweiterungen der GSN entwickelt. Zum einen kann die GSN Pattern-Extensions unterstützen. Pattern-Extensions sind eine Abstraktion einzelner Entitäten oder der Struktur der Notation. Unterschieden wird hierbei zwischen Klassen und Instanzen der jeweiligen Objekte. Die Abstraktion der Struktur kann außerdem helfen um "one-to-one" und "one-to-many" Relationen zwischen Komponenten darzustellen, das dem Betrachter beim Verständnis helfen kann bzw. die Details und Strukturen eines Systems besser erkenntlich macht.

### 2.8.1 Module

Unter gegebenem Ziel, GSN-Diagramme großer Systeme übersichtlicher und verständlicher zu machen, wurde eine Erweiterung entwickelt mithilfe deren man Teilsysteme modularisieren kann. Diese Module sind übergeordnete Elemente, verbunden durch die bekannten Unterstützt-Von sowie Kontext-Von Verknüpfungen, die Teile des Diagramms mit einem Modul-Element substituieren. Werner et al. [2012]

Dafür wird das GSN-Diagramm in zwei Ebenen aufgeteilt. Die Innere Ebene beinhaltet die gewohnte GSN-Struktur. Ein Ziel, verschiedene Teilziele und andere GSN-legale Elemente verknüpft durch Unterstützt-Von sowie Kontext-Von Verbindungen. Diese werden zu einem Modul zusammengefasst.

Ein Modul-Element enthält wie andere GSN-Elemente einen Titel, eine ID sowie eine Beschreibung.

Auf der modularen Ebene sind mit einander verknüpfte Module zu sehen. Aufgebaut wie die innere Ebene, ergibt sich ein wohlgeformtes GSN-Diagramm. Der Unterschied der Ebenen liegt bei den dargestellten Informationen. In der äußeren Ebene, der modularen Ebenen, sind Informationen der Argumentation nicht direkt verfügbar, sondern nur Informationen über den Inhalt einzelner Module, sowie deren Zusammenhang, dargestellt durch bereits bekannte Verknüpfungen. Um Informationen über die Argumentation zu erhalten muss die innere Ebene eines Moduls aufgerufen werden.

Schnittstellen zwischen den einzelnen Modulen können durch Fernziele, Fernkontext, Fernlösungen sowie komplette Modul Referenzen dargestellt werden.

## 2. Hintergrund und theoretisches Vorwissen

Ein **Fern-Ziel** referenziert auf ein Ziel, dass sich in einem anderen Modul befindet.

Ein **Fern-Kontext** Ein Fernkontext referenziert auf ein sich in einem anderen Modul befindlichen Kontext.

Eine **Fernlösung** referenziert einen Beweis oder Beweisschema, dass sich innerhalb eines anderen Moduls befindet.

Eine **Modul Referenz** referenziert auf ein anderes Modul, innerhalb der inneren Ebene eines Moduls.

## 2.9 Zusätzliche Informationen zur GSN-Diagramm Erstellung

Einer der wichtigsten Punkte bei der GSN-Erstellung ist es für den Ersteller des Diagramms Begriffe so zu wählen, dass das Zielpublikum bzw. die Leserschaft versteht was gemeint ist. Der Autor muss sich während der Erstellung in die Rolle des Lesers versetzen.

## 2.10 Evaluation eines GSN-Diagramms

Um ein GSN-Diagramm zu evaluieren kann ein Review durchgeführt werden. Ein Review ist meist die beste Variante um Struktur-, Form- und Syntaxfehler eines GSN-Diagrammes aufzudecken. Auch logische Fehler, mangelhaft abdeckende Beweise und Lücken in der Argumentation spätestens innerhalb eines Reviews sichtbar. Um ein Review erfolgreich durchzuführen sollte man sich an folgende 4 Stichpunkte, die nacheinander abgearbeitet werden können, halten.

- ▷ Verständnis der Argumentation
- ▷ Wohlgeformtheit des Diagramms
- ▷ Aussagekraft
- ▷ Kritik an der Argumentation

**Schritt 1 - Verständnis der Argumentation** Um weitere Schritte zu bearbeiten, ist es erst einmal notwendig, dass der Leser die präsentierte Argumentation versteht. Der Leser muss anhand des gegebenen Diagramms die Behauptungen, Strategien und Annahmen

## 2.10. Evaluation eines GSN-Diagramms

nachvollziehen können. Außerdem sollen die Notationen der einzelnen Elemente überprüft werden.

Gibt es eine andere Dokumentation der Safety-Cases, auf der das GSN-Diagramm aufbaut, so kann anhand dieser verglichen werden, ob alle Elemente der Argumentation übertragen worden sind.

**Schritt 2 - Wohlgeformtheit des Diagramms** Ein GSN-Diagramm wird wohlgeformt bezeichnet sofern keine Struktur- oder Syntaxfehler innerhalb des Diagramms existieren. Ein Strukturfehler ist beispielsweise ein Teilziel, dementsprechend eine Behauptung, für die keine passenden Evidenzen existieren. Ebenso können Beweise deren Rolle in dem Argument unklar sind ein Indiz auf ein Strukturfehler sein. Im Allgemeinen gilt: Ein Diagramm ist wohlstrukturiert sofern es vollständig ist und die Definitionen der Elemente eingehalten wurden.

**Schritt 3 - Aussagekraft** Die Aussagekraft eines GSN-Diagramms wird daran bestimmt, ob alle Elemente ausreichend belegt sind. Des weiteren muss geprüft werden, ob alle Verknüpfungen des Diagramms logisch nachvollziehbar sind.

**Schritt 4 - Kritik an der Argumentation** Im letzten Schritt des Reviews, der Kritik an der Argumentation, geht es darum, ob die Beweise zusammen ausreichend sind um die jeweilige Schlussfolgerung ziehen zu können. Im speziellen soll betrachtet werden, wie direkt, relevant und abdeckend die Beweise sind. Außerdem muss geklärt werden ob die Rahmenbedingungen der Argumentation eingehalten wurden.  
Group et al. [2011]

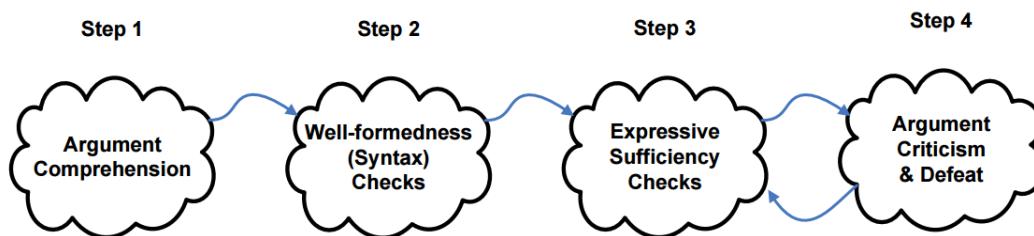


Abbildung 2.13. Schritte der Evaluation eines GSN-Diagramms aus dem GSN Community Standard



# Automatisierbare Eigenschaften

In diesem Kapitel soll es darum gehen, in wie Fern der Benutzer durch Teil- und Vollautomatisierung der Erstellung eines GSN-Diagrammes geleitet und unterstützt werden kann.

Dazu werden zunächst die Richtlinien und Festlegungen des GSN Community Standards analysiert um herauszufinden was ein wohlgeformtes GSN Diagramm ausmacht.

In einem weiteren Schritt sollen die gefundenen Eigenschaften auf ihre Automatisierbarkeit untersucht werden.

## 3.1 Richtlinien und Indizien

Es gilt zunächst zwei unterschiedliche Arten von Richtlinien zu unterscheiden. Zum einen gibt es die technischen Richtlinien, die darauf abzielen ein wohlgeformtes GSN-Diagramm zu zeichnen, diese haben nur sehr bedingt mit dem eigentlichen Inhalt der GSN-Elemente zu tun, sind aber ausschlaggebend für die Darstellung dieser. Zum anderen gibt es die inhaltlichen Richtlinien, die sich ausschließlich mit dem Inhalt der Elemente beschäftigen.

### 3.1.1 Technische Richtlinien

Die technischen Richtlinien beziehen sich auf Einschränkungen und Definitionen der GSN-Elemente. Diese Einschränkungen sind vor allem für die Implementierung relevant.

Zu erst wird anhand der vorliegenden Informationen geklärt werden, wie ein Element aufgebaut ist und welchen Informationen darin enthalten sind.

#### Aufbau eines GSN-Elements

Ein GSN-Element besteht aus folgenden vier Grundkomponenten.

### 3. Automatisierbare Eigenschaften

1. Ein Typ
2. Ein Kürzel
3. Eine Beschreibung
4. Verknüpfungen

Innerhalb der Beschreibung steht die eigentliche Aussage des Elements. Das kann eine Behauptung oder Referenz sein. Der Typ gibt an um was für ein Element es sich handelt. Das Kürzel, im weiteren Verlauf ID genannt, ist eine eindeutige Zuweisung an das Element, das dazu dient alle Elemente auseinander halten zu können. Außerdem muss ein Element Informationen über die Verknüpfungen zu anderen Elementen beinhalten.

### Kategorisierung der Elemente

Alle GSN-Elemente haben eine gewisse Funktion. Diese Funktion variiert von Typ zu Typ. Aus den Funktionen lassen sich gewisse Ähnlichkeiten und Abhängigkeiten lesen, sodass die Elemente kategorisiert werden können. Die Elemente innerhalb einer Kategorie sind nicht identisch, sie besitzen nur Ähnlichkeiten die eine Einstufung möglich macht.

#### 1. Kategorie 1: Ziele

- ▷ Zur Kategorie 1 gehören ausschließlich Ziele bzw. "Top-Goals"
- ▷ Es existiert nur ein Ziel innerhalb eines GSN-Diagramms.
- ▷ Kein Element wird jemals über dem Ziel dargestellt.
- ▷ Ziele besitzen keine eingehenden Verknüpfungen

#### 2. Kategorie 2: Hauptelemente

- ▷ Zur Kategorie Hauptelemente gehören alle Teilziele und Strategien.
- ▷ Jedes Hauptelement ausschließlich Ziel von "Unterstützt-Von" Verknüpfungen.
- ▷ Jedes Hauptelement ist mindestens Ziel einer "Unterstützt-Von" Verknüpfung.

#### 3. Kategorie 3: Nebenelemente

- ▷ Zur Kategorie Nebenelemente gehören Kontexte, Annahmen und Rechtfertigungen.
- ▷ Jedes Nebenelement ausschließlich Ziel von "Kontext-Von" Verknüpfungen.
- ▷ Jedes Nebenelement mindestens Ziel einer "Kontext-Von" Verknüpfung.

#### 4. Kategorie 4: Lösungen

### 3.1. Richtlinien und Indizien

- ▷ Zur Kategorie 4 gehören ausschließlich Lösungen.
- ▷ Eine Lösung besitzt mindestens eine "Unterstützt" Verknüpfung.
- ▷ Eine Lösung keine "Kontext-Von" Verknüpfung.

#### 3.1.2 Inhaltliche Richtlinien

Die inhaltlichen Richtlinien beziehen auf Vorgaben der Beschreibungen innerhalb der Elemente der GSN.

Die Goal Structuring Notation ist grundsätzlich darauf ausgelegt Argumentationen darzustellen. Für die Konstruktion eines GSN-Diagrammes bedeutet das, dass Beschreibungen innerhalb der Elemente verständlich sein müssen. Dafür werden spezielle sprachliche Richtlinien im GSN Community standard genannt. Darunter werden speziell der Satzbau und die Grammatik innerhalb der GSN angesprochen. Satzbau und Grammatik sind nicht nur wegen dem allgemeinen Verständnis bzw. der Qualität der Sprache wichtig, sondern können schlecht gestellte Behauptungen aufdecken, durch eine quantitative Analyse des Satzes. So zum Beispiel sind mehrere Verben innerhalb der Beschreibung eines Ziels oder Teilziels meist ein Zeichen dafür, mehr als eine Behauptung genannt zu haben. Ebenso sind mehrere Substantive innerhalb von Lösungen und Kontexten ein Indiz dafür mehr als eine Referenz zu beinhalten. Group et al. [2011]

#### Die Menge der Informationen

Die Frage nach der Menge der Informationen, die innerhalb eines Elements der GSN sowie über die Hierarchietiefe hinweg abgebildet werden, lässt sich nicht direkt aus dem GSN-Community Standard entnehmen. Es gibt aber verschiedene Indizien darauf.

In dem GSN Community Standard wird erwähnt, dass der Autor unter allen Umständen vermeiden muss, zu viele Informationen innerhalb eines Elements darzustellen. Durch Überladung der Elemente wird nicht nur die Übersichtlichkeit des Diagramms eingeschränkt, sondern oft, ob bewusst oder unbewusst, auch mehrere Aussagen innerhalb eines Elements gemacht.

Die Informationen innerhalb eines Elements sollen so kurz wie möglich gehalten werden, ohne dabei das Verständnis zu gefährden. Ausnahmen sind die Strategie, Annahme und Rechtfertigung, diese müssen mit einem gewissen Detailgrad geschrieben werden, sodass der Nutzer das Element und daraus resultierend die Struktur des Argumentes verstehen kann.

### 3. Automatisierbare Eigenschaften

Um die Informationen möglichst kurz und dennoch aussagekräftig zu schreiben soll der Autor folgende Punkte beachten. Group et al. [2011]

Die **Atomarität**. Jedes Element soll exakt eine Behauptung oder Referenz beinhalten. Mehrere Verben innerhalb eines Ziels können, wie bereits im zu vorigen Abschnitt erwähnt, bedeuten, dass das Ziel mehrere Behauptungen enthält. Innerhalb von Lösungen und Kontexten sind mehrere Substantive meist ein Indikator für mehr als eine Referenz. Group et al. [2011]

Die **Relation zwischen Text und Funktion**. Es ist wichtig dass der Text eines Elements der Funktion des Elements entspricht. Ziele enthalten Behauptungen, Lösungen enthalten Referenzen und Strategien fassen nur das Vorgehen zur Aufspaltung von Zielen auf. Group et al. [2011]

Lösungen und Kontexte beinhalten meist Referenzen auf Dokumente, die sich nicht innerhalb der GSN befinden. Es ist nicht notwendig den Inhalt der Dokumente wieder zu geben, oder zusammenzufassen. Group et al. [2011]

Wichtiger als die Menge der Informationen, ist, dass die Argumentation klar bleibt. Der Text innerhalb eines Argumentes darf in keinem Fall zweideutig sein und alle zum Verständnis notwendigen Informationen müssen in jedem Fall gegeben sein.

Für die Automatisierung heißt das jedoch auch, dass die Informationsmenge kein Indiz auf die Qualität der Informationen sein muss. Also wäre die Einschränkung der Informationsmenge innerhalb eines Elements auf einen Satz, oder eine gewisse Anzahl an Zeichen, nicht produktiv.

Zu der Menge der Informationen über die Hierarchietiefe hinweg gibt es ebenfalls Indizien. Es müssen alle Informationen die Erkenntnis über den Verlauf der Argumentation liefern, gegeben werden. Das heißt aber ebenso, dass die Menge der Informationen innerhalb des Diagramms nicht mit der Qualität des Inhalts in Verbindung gebracht werden kann.

## 3.2 Automatisierungen

Die Automatisierungen basieren auf bereits gefundenen Eigenschaften, können aber auch aus der Implementierung resultieren. Auch Datenprüfungen bzw. Datenvalidierungen werden eingeschlossen.

Folgende Automatisierungen, die auf den technischen Richtlinien basieren, sollen implementiert werden:

### 3.2. Automatisierungen

- ▷ Jede ID ist einzigartig.
- ▷ Jedes Element hat genau einen Typ
- ▷ Jedes Element hat eine Beschreibung
- ▷ Ein Element kann immer nur ein Typ von Verknüpfungen besitzen.
- ▷ Das Haupt Ziel hat keine ausgehenden Verknüpfungen.

Andere Eigenschaften wurden zwar in Erwägung gezogen aber letztendlich nicht als hilfreich befunden. Darunter beispielsweise, dass nur ein Ziel bzw. Goal innerhalb der GSN existiert. Der Standard differenziert zwischen Zielen und Teilzielen, in der Industrie und praktischen Anwendung der GSN wird dies aber häufig außer acht gelassen. Deshalb soll die Verwendung der Ziele und Teilziele optional bleiben und dem Benutzer überlassen werden. Group et al. [2011]

Daraus resultiert aber auch, dass entgegen in Kapitel 3 gemachter Aussagen, Elemente über Zielen dargestellt werden können. Auch dieser Punkt wird deshalb nur bedingt implementiert. Innerhalb der Darstellung wird zuerst die XML-Datei auf ein Ziel durchsucht mit der ID "G1". Dieses Ziel ist das Hauptziel, das eigentliche Ziel und dieses wird ohne Ausnahmen als erstes Element dargestellt.

Außerdem werden zusätzlich noch folgende inhaltlichen Richtlinien und deren Automatisierung betrachtet.

- ▷ Die Rechtschreibung der Beschreibung
- ▷ Der Satzbau innerhalb der Beschreibung
- ▷ Die Grammatik innerhalb der Beschreibung

Die inhaltlichen Richtlinien sind nur schwer zu automatisieren. In der deutschen Industrie kann nicht pauschal gesagt werden, welche Sprache verwendet wird, ob Deutsch oder Englisch. Es kommt auch oft zu einer Mischung der beiden sofern die deutsche Sprache aber englische Fachausdrücke verwendet werden. Das führt dazu das eine automatische Überprüfung der Rechtschreibung, des Satzbaus und der Grammatik sehr schwer zu implementieren ist. Außerdem müsste dafür auf externe, eventuell kostenpflichtige Bibliotheken zurückgegriffen werden, deshalb wurde im Sinne des Werkzeugs gegen solch einen Implementierungsversuch entschieden.



# Die Implementierung

Im folgenden Kapitel werden die Anforderungen an den praktischen Teil der Arbeit geklärt. Dazu werden verschiedene Varianten und Vorgehensweisen beschrieben und im weiteren die Architektur des Werkzeugs entworfen. Darauf folgt der Abschnitt zur Implementierung des Werkzeugs.

## 4.1 Anforderungen

Die Siemens AG, Corporate Technology - Research and Technology Center nutzt bisher Visio zur grafischen Darstellung von GSN-Diagrammen. Microsoft Visio ist ein Programm zur grafischen Darstellung verschiedener Diagramme und Flowcharts. Es ist zwar ein mächtiges Programm mit vielen Funktionen und verschiedenen Speicherformaten, bietet aber keine Funktionen die im direkten Zusammenhang mit der Goal Structuring Notation stehen. Darunter zum Beispiel die im GSN Community Standard spezifizierten Richtlinien die innerhalb der GSN-Diagramme eingehalten werden müssen.

Die Siemens AG würde deshalb eine Lösung bevorzugen, die diese Richtlinien sowie automatisierbare Eigenschaften umsetzt. Diese automatisierbaren Eigenschaften sollen nach Möglichkeit die Erstellungszeit eines GSN-Diagrammes verkürzen und dessen Qualität verbessern. Außerdem soll nach Möglichkeit ein XML basierter Ansatz, der sich in einem Browser öffnen lässt, gewählt werden. Die Darstellung soll dem Modellierer automatisch bei der Erstellung eines GSN-Diagrammes unterstützen. Des weiteren sollen nach Möglichkeit weitere Automatisierungen implementiert werden, die den Nutzer bei der GSN-Diagramm Erstellung unterstützen.

Die entstehende Lösung soll im speziellen auf den Betriebssystemen Windows 7 und Windows 8.1 sowie Windows 10 lauffähig sein.

#### 4. Die Implementierung

### 4.2 Entwurf - Teil 1 - Speicherung der Daten

Anhand der Anforderungen soll nun die Architektur der Anwendung erstellt werden. Die Darstellung soll, wie bereits in den Anforderungen genannt, auf einer XML-Basis entstehen. Innerhalb der XML-Datei sollen sich alle Informationen, die zum Zeichnen einer wohlgeformten GSN-Struktur notwendig sind, befinden. Die XML-Datei soll auf einem Schema erstellt werden, sodass gewillte Personen die XML-Datei direkt bearbeiten können. Um das zu ermöglichen muss die Struktur des GSN-Diagrammes leicht Verständlich sowie einfach zu Erstellen sein. Es soll möglich werden mit geringer Einarbeitungszeit eine XML-Datei dieses Schemas schreiben zu können. ?

#### 4.2.1 Schema der XML Datei

Die XML besitzt typischer Weise eine Kopfzeile in der die genutzte XML Version, die Textkodierung sowie die Verwendung einer DTD innerhalb des Standalone Tags. Auf die Verwendung einer DTD wird verzichtet. Im weiteren folgt die Stylesheet Definition. Der Typ des Stylesheets und der Ort an dem sich die Stylesheet Datei befindet.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<?xml-stylesheet type="text/xsl" href="gsn.xsl"?>
```

Um das XML-Schema möglichst einfach zu halten sollen die einzelnen Elemente der GSN innerhalb von Element-Blöcken beschrieben werden. Ein Element-Block ist ein XML-typischer Tag der ein Element umrandet. Im folgenden Listing wird der Sachverhalt beschrieben.

```
<elements>
  <element>
    <information1>Erstes Element</information1>
    <information2></information2>
  </element>
  <element>
    <information1>Zweites Element</information1>
    <information2></information2>
  </element>
</elements>
```

Zunächst muss sichergestellt werden welche Informationen festzuhalten werden müssen. Dies wurde bereits in den vorherigen Kapiteln angesprochen.

## 4.2. Entwurf - Teil 1 - Speicherung der Daten

Im folgenden werden die Bestandteile eines Elements der GSN aufgelistet und daraufhin im Detail beschrieben.

1. Ein Typ
2. Eine ID
3. Die Beschreibung
4. Die Verknüpfungen

### Der Typ

Der Typ eines Elementes gibt an, um welche Art von Element es sich handelt. Ein Typ kann dementsprechend mehrfach vergeben werden. Der Typ ist, wie im folgenden Kapitel Verknüpfungen beschrieben, ausschlaggebend für die ausgehenden Verknüpfungen.

### Die ID

Um ein Element eindeutig von anderen Elementen unterscheiden zu können, muss eine ID zugewiesen werden. Die ID zu einem Element ist einzigartig, das heißt Jede ID kann nur einmal vergeben werden. Jedes Element besitzt eine ID. Die ID folgt dem Schema des GSN Community Standards, ein Kürzel gefolgt von einer Zahl. Das Kürzel beinhaltet eine Kurzform des Typs des jeweiligen Elements. Die Zahl wird von der jeweiligen Anzahl der Elemente selbigen Typs bestimmt, beispielsweise "C3" ist der dritte erstellte Kontext des Diagramms. Die ID darf aber unter keinen Umständen als Position des Elements fehlinterpretiert werden. Die Position innerhalb des Diagramms bestimmen die jeweiligen Verknüpfungen, nicht die ID. Die ID ist lediglich zur eindeutigen Identifizierung eines Elements ausgelegt.

**Table 4.1.** Tabelle der Kürzel/ID des jeweiligen Typs

Typ (deutsch)	Typ (englisch)	ID
Ziel	Goal	G
Strategie	Strategy	S
Lösung	Solution	Sn
Kontext	Context	C
Rechtfertigung	Justification	J
Annahme	Assumption	A

#### 4. Die Implementierung

##### **Die Beschreibung**

Die Beschreibung ist der Kern eines Elements. In der Beschreibung steht eine Behauptung oder Referenz wie bereits in Kapitel 1 spezifiziert.

##### **Die Verknüpfungen**

Die Verknüpfungen sind die Verbindungen zwischen Elementen, in einem GSN-Diagramm über Kanten dargestellt. Üblicherweise werden die Verknüpfungen von Eltern- zu Kindeselementen innerhalb der Eltern dargestellt. In diesem Fall ist es aber von Vorteil, die Verknüpfungen innerhalb der Kindeselemente zu bestimmen. Der Vorteil besteht darin, dass man zum Zeitpunkt des Erstellens eines Elementes noch nicht wissen muss, welche Elemente noch folgen. Die folgenden Elemente könnten natürlich nachträglich in das Elternelement eingetragen werden. Das ist aber zu umständlich und führt zu einem schlechten Workflow. Deshalb werden die Verknüpfungen innerhalb der Kindeselemente bestimmt, so dass der Ersteller die Elemente nicht mehrfach bearbeiten muss.

Alle Haupt- und Nebenelemente sowie die Lösung haben mindestens eine Verknüpfung. Nebenelemente und Lösungen können mehrere Verknüpfungen besitzen.

### **4.3 Entwurf - Teil 2 - Das Stylesheet**

Die erstellte XML-Datei zur Informationsspeicherung muss nun innerhalb eines Browsers wohlgeformt dargestellt werden.

Es gibt verschiedene Varianten zur Darstellung von XML-Dateien. Eine davon ist ein sogenanntes Stylesheet. Innerhalb der Stylesheets wird ein Vorgehen beschrieben, wie die Daten der XML-Datei im Browser dargestellt werden sollen. Andernfalls können auch externe, teils im Browser aufrufbare Programme, Plugins und Addons dazu verwendet werden um Daten im Browser sichtbar zu machen.

Externe Programme haben den Vorteil, das gezeichnete Diagramme strukturierter und schöner dargestellt werden können. Der Nachteil ist jedoch, dass sie meist eine Vorinstallation erfordern und nicht selten kostenpflichtig sind.

Stylesheets dagegen, können generell ohne Vorinstallation oder Änderung bestehender Daten verwendet werden. Jeder moderner Browser lässt die Darstellung von Daten mittels Stylesheets zu. Aufgrund der allgemeinen Nutzbarkeit von Stylesheets wird diese Variante in diesem Projekt bevorzugt.

Es gibt momentan primär zwei Technologien zur Darstellung für XML basierende Daten innerhalb eines Browsers. Zum einen gibt es Cascading Stylesheets (CSS), zum anderen gibt es Extensible Stylesheet Language (XSL). XSL und CSS bieten sehr ähnliche Funktionen. Im Allgemeinen gilt: CSS bietet die schönere Formatierung, wobei XSL mehr Funktionalität

#### 4.4. Entwurf - Teil 3 - Automatisierte XML Generierung

bietet. Bos et al. [2005] Bray et al. [2011]

CSS bietet bessere Darstellungsmöglichkeiten, hat jedoch den Nachteil, dass die Filterung von Daten nur bedingt möglich ist. Da aber gerade die Filterung der Daten zur Darstellung wohlgeformter GSN-Diagramme essentiell notwendig ist, wird im weiteren Verlauf XSL verwendet. Im speziellen sollen XSL Transformation (XSLT) genutzt werden, um das gegebene XML-File innerhalb eines Browsers zu HTML zu transformieren, woraufhin die HTML Version vom Browser interpretiert und dargestellt wird. sel [2015]

Auch unter den Stylesheets gibt es verschiedene Variationen, intern und extern definierte Stylesheets. Interne Stylesheets werden, wie deren Name schon vermuten lässt, innerhalb der XML-Datei definiert. Das hat den Vorteil, dass zur Darstellung und Speicherung der Informationen nur eine Datei notwendig ist. Der Nachteil ist, dass der Aufbau der XML-Datei durch die zusätzlichen Informationen des Stylesheets wesentlich komplexer wird. Diese Komplexität muss aber vermieden werden, da es aufgrund der Anforderungen möglich bleiben soll, dass der Nutzer die XML-Datei selbst, ohne unterstützende Werkzeuge schreibt. Deshalb wird entschieden ein externes Stylesheet zu verwenden. Dieses soll generell für alle XML-Dateien die dem Schema folgen verwendbar sein.

#### 4.4 Entwurf - Teil 3 - Automatisierte XML Generierung

Die bisher erstellten Komponenten sind zur Speicherung und Darstellung von Daten in GSN-Diagrammen geeignet, jedoch nicht zur Automatisierung des Erstellungsverfahrens sowie der Einhaltung der Richtlinien. Die Automatisierungen sollen in einem weiteren Werkzeug umgesetzt werden, einem GSN-Generator. Dieser GSN-Generator hat primär zwei Aufgaben. Zum einen soll es möglich sein auf Eingaben des Nutzers eine XML-Datei im passenden Schema automatisch generieren zu lassen. Zum anderen soll der XML-Generator auch die Eingaben des Nutzers mit den Richtlinien und Vorgaben des GSN Community Standards abgleichen. Im Fall eines Verstoßes gegen diese Richtlinien soll eingegriffen werden. Außerdem sollen illegale Datenzustände vermieden werden. Speziell die Eingabe und das Laden von XML-Dateien sollen überprüft werden um illegale Datenzustände zu verhindern. Des weiteren soll die Darstellung dem Modellierer dabei unterstützen, ein wohlgeformtes GSN-Diagramm zu erstellen. Das wird durch eine weitere tabellarische Darstellung innerhalb des GSN-Generators bewerkstelligt. Anhand dieser Tabelle soll der Nutzer die bereits eingegebenen Daten einsehen und ändern können.

#### 4.5 Übersicht der Architektur

Der Aufbau und die Komposition einzelner Komponenten der Architektur soll anhand folgendem Diagramm gezeigt werden. Im speziellen soll das Diagramm zeigen, welche

## 4. Die Implementierung

Komponenten verwendet wurden und wie diese miteinander interagieren.

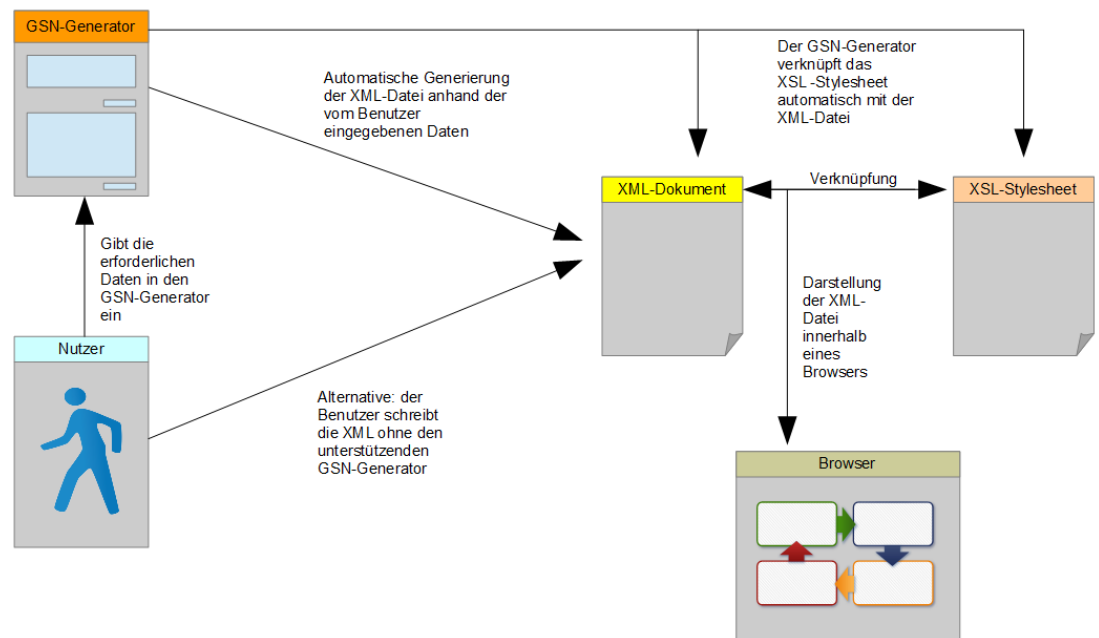


Abbildung 4.1. Darstellung der verwendeten Komponenten

## 4.6 Implementierung

Im folgenden Kapitel wird die Umsetzung der drei Komponenten beschrieben.

### 4.6.1 Darstellung der XML-Datei

Die grafische Abbildung innerhalb des Browsers basiert auf einem tabellarischen Ansatz. Rekursiv werden für alle Hauptelemente Tabellen erstellt, diese Tabellen beinhalten das Hauptelement, auf dem sie aufbauen, sowie alle kontextuellen Bedingungen die für das Element gelten, somit alle Nebenelemente die in Verbindung mit diesem Hauptelement stehen. Außerdem werden alle Hauptelemente darunter gelegenen Ebenen mit selbigem Tabellenrahmen umrandet. Das soll dazu führen, dass die Zugehörigkeit der Elemente sichtbar wird, speziell die Vater-Kind Beziehung.

Des weiteren wird ein Farbschema angewandt, dass die verschiedenen Elemente von

einander Unterscheidbar machen soll.

#### 4.6.2 Struktur der XML-Datei

Die Struktur der XML-Datei soll anhand eines Beispiels aus dem GSN Community Standard gezeigt werden.

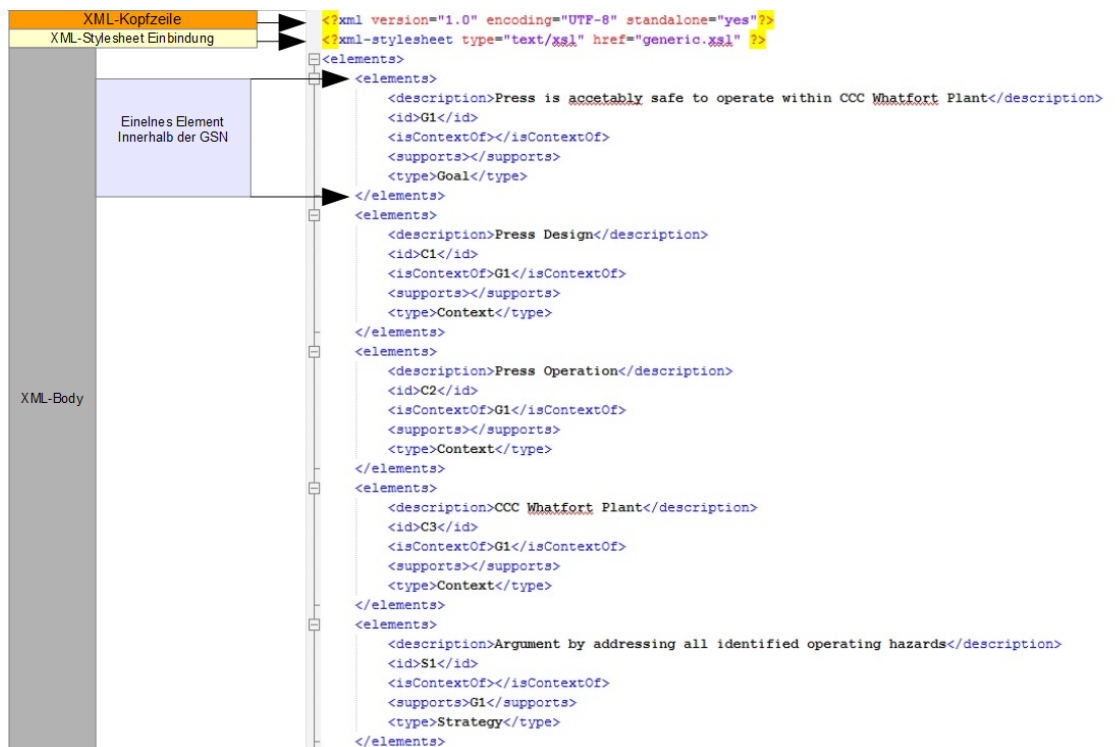


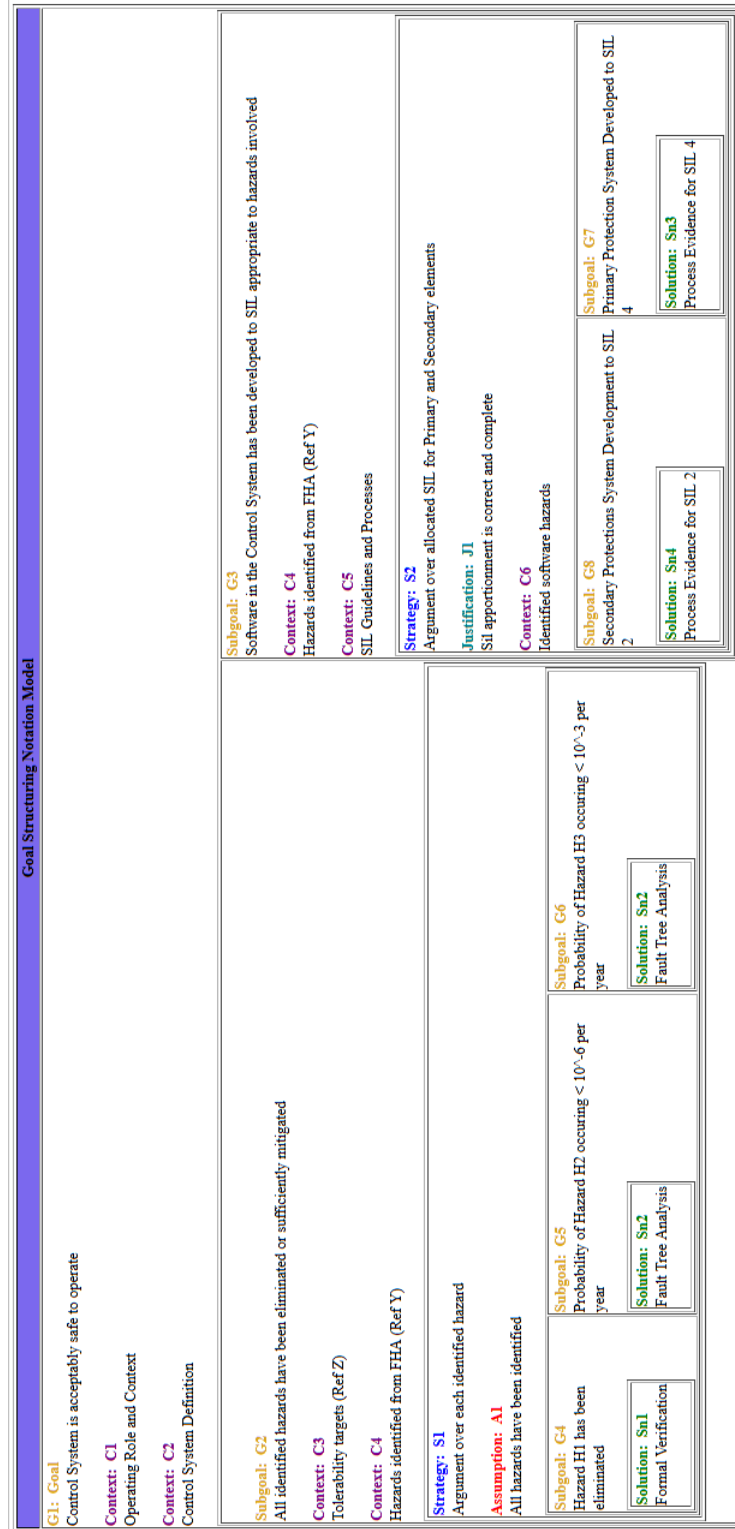
Abbildung 4.3. Beispiel der Struktur einer XML-Datei passenden Schemas

Die Kopfzeile beinhaltet die XML-Version, das Encoding der Schriftart sowie den Tag Standalone, mithilfe dessen man ein DTD-Schema mit der XML koppeln kann. Daraufhin folgt die Einbindung des Stylesheets. Diese beinhaltet den Typ des Stylesheets, beispielsweise CSS oder XSL. Darauf folgt der Programmpfad, der Ort, an dem das Stylesheet zu finden ist.

Daraufhin kommt der XML-Body, in dem die Informationen des darzustellenden GSN-Diagramms gesichert werden. Innerhalb des XML-Bodies befinden sich die Strukturen der

#### 4. Die Implementierung

38



**Abbildung 4.2.** Darstellung mit eingebundenem Stylesheet. Beispiel entnommen aus dem GSN Community Standard

einzelnen Elemente.

### 4.7 Aufbau der XML

Die Extensible Markup Language (kurz XML) Datei enthält die inhaltlichen Informationen, die zum Zeichnen des GSN-Diagramms notwendig sind. Diese kann vom GSN-Generator produziert werden und folgt einem im Verlauf der Sektion beschriebenen Schema. Dieses Schema kann auch ohne Tool in einem beliebigen Editor, beispielsweise Notepad++, geschrieben werden. Sofern das Schema der Datei eingehalten wird.

Das Schreiben der XML von Hand hat jedoch den Nachteil, dass es keine Sicherheitsprüfung gibt ob das Diagramm wohlgeformt und korrekt dargestellt wird, wie vom Ersteller geplant.

#### 4.7.1 ID

Jedes Element hat eine einzigartige ID Festgehalten in folgender Struktur:

#### 4.7.2 Typ

Der Typ ist notwendig um eine automatisierbaren Eigenschaften zu implementieren.

#### 4.7.3 Beschreibung

Die Beschreibung enthält die Informationen, die innerhalb der GSN-Komponente jeweiligen Typs dargestellt werden sollen.

#### 4.7.4 Verknüpfungen

Anders als im GSN-Community Standard spezifiziert werden Verknüpfungen jeglicher Art im Kindes Knoten statt dem Eltern Knoten festgehalten. Das hat den Vorteil, das zum Zeitpunkt des Erstellens des Elternelements kein Wissen über zukünftige ausgehende Kanten notwendig sind. Um die Verknüpfungen ohne Abfragen, zugunsten des Aufbaus des Stylesheets, von einander Trennen zu können werden diese in zwei verschiedenen Tags festgelegt.

#### 4. Die Implementierung

##### **Unterstützt**

Die Unterstützt Verknüpfung verknüpft alle Hauptelemente und Lösungen mit einem Hauptelement oder Ziel darüber gelegener Ebene.

##### **Ist-Kontext-Von**

Die Ist-Kontext-Von Verknüpfung verknüpft alle Nebenelemente mit dem Ziel und deren Hauptelementen.

#### **4.7.5 Ein Element der GSN innerhalb der XML-Datei**

Im folgenden soll der Aufbau eines XML-Elements veranschaulicht werden.

```
<elements>
  <description> Fault Tree Analysis </description>
  <id>Sn2</id>
  <isContextOf></isContextOf>
  <supports>G5, G6</supports>
  <type>Solution</type>
</elements>
```

Im dargestellten Beispiel handelt es sich um eine Lösung, diese Lösung besteht wie schon zuvor beschrieben aus einer Beschreibung, einer ID, den Verknüpfungen sowie eines Typs. Sollte das Element mehr als eine Verknüpfung besitzen, so wird innerhalb des Tags mit Komma und Leerzeichen getrennt, wie im vorherigen Listing beispielhaft gezeigt wurde.

#### **4.8 Der GSN-Generator**

Der GSN-Demonstrator wird aufgrund von vorhandenen Vorkenntnissen in Java implementiert. Der Vorteil an Java liegt an dessen universellen Nutzbarkeit und vielen schon vorhandenen Bibliotheken, deren Nutzung auf kostenfreien Lizenzen basiert. Außerdem ist zur Ausführung des Werkzeugs keine Installation erforderlich, lediglich Java muss installiert sein.

#### **4.9 verwendete Technologien**

Im folgenden werden die verwendeten Technologien aufgelistet und daraufhin Beschrieben.

- ▷ Java SE Development Kit 8
- ▷ Java Architecture for XML Binding

Java SE Development Kit 8 (JDK 1.8) ist die momentan aktuelle Entwicklungsplattform für Java. Das Entwicklungs-Kit bietet eine Ansammlung von Funktionen und Methoden, die dem Entwickler bei der Implementierung helfen.

Java Architecture for XML Binding (JAXB) ist eine Java Bibliothek mithilfe der XML Elemente zu Java Objekten transformiert werden können. Ebenso können Java Objekte zu XML Elementen transformiert werden. Durch die Verwendung von JAXB muss das Schema der XML einem gewissen Aufbau folgen. Außerdem ist es beim Importieren wichtig genau dieses Schema zu verwenden, sollte man die XML-Datei ohne unterstützendes Werkzeug geschrieben haben.

Durch JAXB kann ein automatisierter Im- und Export einer XML-Datei gewährleistet werden.

### 4.10 Layout des GSN-Generators

Das Layout bzw. der Aufbau des GSN-Generators soll möglichst minimalistisch sein, um möglichst geradlinig damit arbeiten zu können. Der GSN-Generator besteht aus verschiedenen Feldern, die zur Eingabe gedacht sind dazu kommen noch Buttons zum Laden, also dem Import einer XML-Datei, dem Add Button, dieser erstellt ein Element anhand der eingegebenen Daten und verwaltet diese innerhalb einer Tabelle und einem SAVE-Button der dazu gedacht ist ein XML-File zu generieren und an einem beliebigen Ort abzuspeichern. Die GUI wurde mit Java Swing Komponenten realisiert.

#### 4. Die Implementierung

**GsnGenerator**

This is a Tool to create an XML document for Goal Structuring Notation Diagrams Load

Type: Subgoal

ID/Headline: G4

Description:

Hazard H1 has been eliminated

Supports: S1 Is-context-of:  Add

Type	ID	Description	Supports	IsContextOf
Goal	G1	Control System is acceptably safe to operate		
Context	C1	Operating Role and Context		G1
Context	C2	Control System Definition		G1
Subgoal	G2	All identified hazards have been eliminated or sufficiently mit...	G1	
Subgoal	G3	Software in the Control System has been developed to SIL a...	G1	
Context	C3	Tolerability targets (Ref Z)		G2
Context	C4	Hazards identified from FHA (Ref Y)		G2, G3
Context	C5	SIL Guidelines and Processes		G3
Strategy	S1	Argument over each identified hazard	G2	
Strategy	S2	Argument over allocated SIL for Primary and Secondary ele...	G3	
Assumption	A1	All hazards have been identified		S1
Justification	J1	Sil apportionment is correct and complete		S2
Context	C6	Identified software hazards		S2
Subgoal	G5	Probability of Hazard H2 occuring < 10 <sup>-6</sup> per year	S1	

Generates the XML document and attaches it to an XSL document for displaying purpose Save

Abbildung 4.4. Layout des GSN-Generators

# Evaluation

Im Kapitel der Evaluation werden der GSN-Generator, die Struktur der entstehenden XML-Datei sowie die Darstellung der GSN-Diagramme im Browser beurteilt. Dazu wird dieses Kapitel in drei Teile aufgeteilt. Als erstes sollen die implementierten Automatisierungen evaluiert werden. Danach folgt eine Sektion zu den Tests, mit welchen die drei Komponenten getestet wurden. Darauf hin soll im speziellen die Frage nach deren Benutzbarkeit geklärt werden. Im Rahmen der Evaluation wird dazu eine Fall- und Nutzerstudie durchgeführt.

## 5.1 der Automatisierungen

Ziel der Arbeit ist es, ein teil automatisiertes Werkzeug zu implementieren. Im folgenden werden die implementierten Automatisierungen noch einmal aufgelistet.

### 5.1.1 Funktionale Automatisierungen

Eine Automatische Generierung einer XML-Datei passenden Schemas wurde mithilfe des GSN-Generators umgesetzt. Zu der bereits genannten Generierung, dem Export, wurde auch ein passender, automatisierter Import implementiert. Die Verknüpfung der XML-Datei mit dessen XSL-Stylesheet wurde ebenfalls automatisiert. Nicht automatisiert wurde jedoch die Erstellung des Stylesheets, dieses Stylesheet muss manuell in den jeweiligen Ordner, in dem sich die XML-Datei befindet, gezogen werden.

### 5.1.2 Automatisierungen der technischen Richtlinien

Im folgenden werden die gefundenen Automatisierungen der technischen Richtlinien noch einmal genannt.

- ▷ Jede ID ist einzigartig.
- ▷ Jedes Element hat genau einen Typ
- ▷ Jedes Element hat eine Beschreibung

## 5. Evaluation

- ▷ Ein Element kann immer nur ein Typ von Verknüpfungen besitzen.
- ▷ Das Haupt Ziel hat keine ausgehenden Verknüpfungen.

Die ausgewählten technischen Richtlinien wurden so implementiert, dass die Daten nach der Erstellung eines Elementes validiert werden. Das geschieht vor dem Eintrag in die tabellarische Ansicht innerhalb des Generators. Werden ein oder mehrere Verstöße gegen die technischen Richtlinien gefunden, wird eine Liste an Fehlern ausgegeben, in der alle Verstöße aufgelistet sind.

Selbiges gilt für den Import der Dateien. Um illegale Datenzustände zu vermeiden wird auch während des Imports validiert ob die eingegebenen Dateien gegen die Richtlinien verstoßen. Falls ein oder mehrere Verstöße vorliegen, so wird analog zum manuellen Eingeben von Daten, eine Fehlerliste mit den Verstößen ausgegeben.

### 5.1.3 Automatisierungen der inhaltlichen Richtlinien

Auf die Implementierung der inhaltlichen Richtlinien wurde, wie bereits in Kapitel 3 erwähnt, aufgrund von sehr hohem Implementierungsaufwand und sprachlicher Komplikationen verzichtet. Diese liegen somit in der Verantwortung des Autors.

## 5.2 Tests

Um die Qualität des Codes und der Funktionsweise der einzelnen Komponenten sicherzustellen, werden verschiedene Tests durchgeführt.

Die XML-Datei folgt einem vordefinierten Schema und erfordert somit keine direkten Tests.

Das Stylesheet bzw. die XSL-Datei wird während des Entwicklungsprozesses mit verschiedenen XML-Beispieldateien getestet. Innerhalb dieser Beispieldaten wird versucht, alle Fälle der GSN-Anordnung abzudecken.

Der GSN-Generator wird ebenfalls mit den erstellten XML-Beispieldaten getestet. Anhand von Tests mit Beispieldaten wurde die Kompatibilität auf den in den Anforderungen genannten Betriebssystemen getestet. Das entstandene Werkzeug ist, sofern Java auf dem System installiert ist, problemlos auf den Betriebssystemen Windows 7 und Windows 8.1 ausführbar. Zudem werden noch automatisierte JUnit-Tests verwendet um einzelne Komponenten innerhalb des Generators zu testen.

Die Junit-Tests sind speziell darauf ausgelegt einzelne Methoden zu testen und deren Funktion sicherzustellen. Im speziellen wird die Logik hinter dem Programm getestet werden. Dazu werden für Listenoperationen und die XML-File Generierung mit JUnit-Tests versehen. Im speziellen wird auch die Validierung der Daten getestet. Die automatisierten

Tests wurden erfolgreich durchgeführt.

Auf automatisierte Tests der GUI wurde aufgrund von einem sehr hohen Implementierungsaufwand und geringem Nutzen verzichtet. Die GUI wurde manuell getestet, was aufgrund der geringen Anzahl an GUI-Elementen problemlos im Bereich des machbaren liegt.

### 5.3 Aufbau der Studie

Die Studie soll Aspekte von Nutzerstudien sowie Fallstudien vereinigen. Dazu wird ein kleines Teilnehmer Umfeld zur Umsetzung der Tools befragt. Die Befragung wird mithilfe eines Fragebogens durchgeführt. In diesem sind Fragen zu folgenden drei Kategorien zu finden:

1. Die Darstellung
2. Der Aufbau der XML-Datei
3. Der GSN-Generator

#### 5.3.1 Kategorie: Darstellung

Um die Befragung zur Darstellung aussagekräftig zu gestalten, soll sich anhand von Beispielen zeigen, ob der implementierte Lösungsansatz der Vorstellung der Teilnehmer entspricht. Um eine möglichst praxisrelevante Umgebung zu schaffen werden Beispieldaten direkt aus dem GSN Community Standard entnommen und in zwei Varianten präsentiert.

Zum einen wird eine aus dem GSN Community Standard stammende, gezeichnete Lösung bereitgestellt. Zum anderen eine Lösung, die mithilfe der entstandenen Tools umgesetzt ist. Anhand des direkten Vergleichs sollen sich Vor- und Nachteile der beiden Lösungen heraus kristallisieren.

#### 5.3.2 Kategorie: Aufbau der XML-Datei

Um den Aufbau der XML-Datei zu evaluieren wird die schon in der Darstellung verwendete XML-Datei wiederverwendet. Der Teilnehmer soll anhand der gegebenen Datei die Struktur der XML-Datei bewerten. Aufgrund der Anforderungen wird Wert darauf gelegt, ob der Nutzer eine Datei dieses Formats mit geringem Zeitaufwand selbst schreiben kann. Im Weiteren soll gefragt werden, wie hoch der Teilnehmer den Zeitaufwand zur

## 5. Evaluation

Einarbeitung in die XML-Struktur schätzt.

### 5.3.3 Kategorie: GSN-Generator

Um den GSN-Generator zu evaluieren soll dieser in zwei Vergleichen dargestellt werden. Zum einen sollen Vor- und Nachteile des GSN-Generators gegenüber der Methode ohne GSN-Generator bewertet werden. Andererseits soll aber auch die Kombination der drei erstellten und zusammenhängenden Komponenten bewertet werden.

## 5.4 Hypothesen der Studie

Die Durchführung der Fall- und Nutzerstudie basiert auf folgenden Hypothesen.

### Hypothesen zur Darstellung

1. Der Nutzer kann sich an eine tabellarische Darstellung innerhalb des Browsers gewöhnen.
2. Die tabellarische Darstellung ist der eines gezeichneten Diagrammes ebenbürtig.
3. Einzelne Elemente der Goal Structuring Notation sind gut voneinander unterscheidbar

### Hypothesen zur XML-Datei

1. Um eine XML-Datei passenden Schemas schreiben zu können, wird eine sehr geringe Einarbeitungszeit benötigt.
2. Die Erstellung einer XML-Datei passenden Schemas kann ohne unterstützendes Werkzeug durchgeführt werden
3. Die Erstellungszeit einer XML-Datei ist geringer als eines zu Zeichnen.

### Hypothesen zum GSN-Generator

1. der Aufbau des GSN-Generators ist logisch und strukturiert.
2. der Zweck und Funktionsweise des GSN-Generators wird durch seinen Aufbau vermittelt.
3. Die Erstellung einer XML-Datei mithilfe des GSN-Generators ist schneller als ohne diesen

## 5.5. Durchführung der Studie

4. Die Erstellung eines GSN-Diagrammes mithilfe aller Komponenten ist schneller als ein GSN-Diagramm zu zeichnen.

Die Studie soll zeigen, ob die genannten Hypothesen belegt werden können oder nicht. Dadurch soll sich die Nutzbarkeit der Werkzeuge beurteilen lassen sowie etwaige Änderungen in Betracht ziehen.

## 5.5 Durchführung der Studie

Die Studie wird mittels Online-Fragebogen über Google Forms durchgeführt. Die Teilnehmer erhalten einen Link zum Fragebogen. Im Fragebogen selbst gibt es einen weiteren Link zum herunterladen der Beispieldateien. Daraufhin wird der Teilnehmer gebeten die gegebenen Fragen abzuarbeiten.

### 5.5.1 Allgemeines zur Studie

Die Studie wird in einem kleinen Umfeld und daher geringer Teilnehmerzahl durchgeführt. Die Teilnehmer sind zum einen Personal der Siemens AG, Corporate Technology - Research and Technology Center, zum anderen Softwaretechnik Studenten der Universität Stuttgart. Durch die geringe Teilnehmerzahl sind die Resultate der Studie nicht repräsentativ für die Allgemeinheit, dennoch lassen sich aus den Resultaten gewisse Tendenzen herauslesen die vor allem zur Verbesserung und Qualitätssicherung der entstandenen Werkzeuge beitragen sollen.

## 5.6 Auswertung der Studie

Im folgenden wird die Studie anhand der genannten Hypothesen ausgewertet.

Die Mehrheit der Teilnehmer ist überzeugt, dass die Darstellung nicht deutlich unübersichtlicher als die eines gezeichneten GSN-Diagrammes ist. Außerdem sind sich die meisten Teilnehmer einig, dass sie sich an eine schlichtere tabellarische Lösung durchaus gewöhnen könnten.

Einzelne Komponenten des Diagrammes sind mäßig voneinander Unterscheidbar, dies könnte in zukünftiger Arbeit noch verbessert werden, in dem Nebenelemente nicht nur Farblich von deren zugehörigen Hauptelementen getrennt werden sondern auch eigene Rahmen erhalten.

## 5. Evaluation

Das Farbschema der Darstellung wurde von der Mehrheit der Teilnehmer als akzeptabel befunden. Jedoch gibt es Teilnehmer die sich ein dezenteres Farbschema wünschen würden.

Die Meinungen zu der Struktur der XML-Datei sind positiv, aber durchwachsen. Die Einarbeitungszeit in eine XML-Datei dieses Schemas ist aber von allen Beteiligten als gering eingestuft worden.

Bei der Frage, ob eine XML-Datei passenden Schemas ohne GSN-Generator schneller zu erstellen sei als eines zu zeichnen wurden sehr unterschiedliche Antworten gegeben. Die durchwachsenen Bewertungen führen zu dem Schluss, dass die Affinität zu XML bzw. Beschreibungssprachen generell sehr variiert. Es liegt in der Präferenz des Anwenders, ob er lieber eine XML-Datei schreibt oder ein Diagramm zeichnet. Es lässt sich hier lediglich sagen dass die Mehrheit es bevorzugen würde ein GSN-Diagramm zu zeichnen als eine XML-Datei passenden Schemas selbst zu schreiben.

Auf die Frage, ob man dem GSN-Generator seine Funktion ansieht, haben die Teilnehmer sehr verschieden geantwortet. Das Layout des GSN-Generators wurde aber einstimmig als gut befunden. Auch die Handlichkeit des GSN-Generators wurde von der Mehrzahl der Teilnehmer als intuitiv bewertet.

Alle Beteiligten sind sich einig, dass die Erstellung der XML-Datei mithilfe des GSN-Generators weniger Zeit in Anspruch nimmt als ohne diesen. Ebenso ist die Mehrzahl davon überzeugt, dass sich ein GSN-Diagramm mithilfe der drei implementierten Komponenten schneller Erstellen lässt, als eines zu Zeichnen

### 5.6.1 Fazit

Die Auswertung der Studie führt zu dem Schluss, dass die Anwendung den Erwartungen und Anforderungen gerecht wird. Abgesehen von der Hypothese, dass sich eine XML-Datei schneller schreiben lässt, als ein GSN-Diagramm zu zeichnen, wurden alle Hypothesen von der Mehrzahl der Teilnehmer bestätigt.

Das Programm erfüllt seinen Zweck und seine Daseinsberechtigung, könnte aber durch verschiedene Änderungen noch verbessert werden. Die gewünschten Änderungen sind verschiedener Art und widersprechen sich teilweise. Dies ist vor allem beim Farbschema der Darstellung zu erkennen. Um allen Benutzern eine Ansicht ihrer Wahl bieten zu können, wäre es möglich in Zukunft verschiedene Farbschemata der Darstellung bereit zu stellen.

Verbessern könnte man zudem auch die Darstellung der Abhängigkeiten einzelner Komponenten. Das könnte unter anderem über eine bessere Kapselung mittels eigener Rahmen

## 5.6. Auswertung der Studie

oder anderer Farblicher Einstufung geschehen.

Im weiteren könnte die Darstellung erweitert werden um ein GSN-Diagramm anhand der in der XML-Datei gegebenen Informationen zu zeichnen. Ein automatisiertes Verfahren um ein GSN-Diagramm zu zeichnen wäre wohl die einfachste Methode, um alle Teilnehmer zufrieden zu stellen. Der Implementierungsaufwand einer solchen Lösung ohne vorhandene Bibliotheken ist jedoch sehr hoch.

Im Großen und Ganzen lässt sich aber sagen, dass die Ziele des praktischen Teils der Arbeit erreicht wurden. Das Werkzeug übertrifft die bisherige Lösung in den Punkten Automatisierung, Erstellungszeit und Unterstützung des Anwenders. Dass das GSN-Diagramm nicht wie im GSN Community Standard gezeichnet, sondern alternativ dargestellt wird, wurde von den Teilnehmern unterschiedlich bewertet. Um alle Teilnehmer zufrieden zu stellen wäre allerdings die gezeichnete Variante die besser.



# Zusammenfassung und Ausblick

In diesem Kapitel befindet sich die Zusammenfassung und der Ausblick.

## 6.1 Zusammenfassung

In der heutigen Industrie wird der Nachweis der Sicherheit eines Systems immer komplexer. Diese Komplexität führt dazu, dass die Dokumentation der Sicherheit ebenfalls komplexer und damit unverständlicher wird. Um das Verständnis zu fördern, wird in der Industrie mittlerweile das Konzept der Safety-Cases verwendet, um Argumentationen über die Sicherheit eines Systems darzustellen und zu dokumentieren. Safety-Cases bestehen aus drei Komponenten, Voraussetzungen und Sicherheitszielen, Beweismaterial und einem Argument, dass die Voraussetzungen und Beweise miteinander verknüpfen soll.

Diese Safety-Cases werden nicht selten innerhalb eines einfachen Textes beschrieben. Diese Texte sind je nach Autor nicht immer wohlstrukturiert und verständlich, daher werden Methoden gesucht, um Safety-Cases besser darstellen zu können.

Die Goal Structuring Notation ist ein Mittel, um Argumentationen grafisch darstellen zu können. Diese grafische Darstellung hat den Vorteil, dass sie auch bei komplexeren Argumentationen übersichtlich und verständlich bleibt. Die GSN hat verschiedene Anwendungsfelder, unter anderem kann die GSN dazu verwendet werden, die zuvor genannten Safety-Cases grafisch und wohlstrukturiert abzubilden.

In dieser Bachelorarbeit wurde ein Ansatz entwickelt, um die Goal Structuring Notation innerhalb eines Browsers darstellen zu können. Zudem wurden Automatisierungen gefunden und implementiert, um dem Benutzer die Erstellung eines GSN-Diagrammes zu erleichtern und diesen bei dabei zu unterstützen. Außerdem werden die vom Benutzer eingegebenen Daten validiert. Das dient dazu, die in den Themen relevanten Standards, speziell dem GSN Community Standard Version 1, genannten Richtlinien einzuhalten und so den Nutzer bei der Erstellung eines wohlstrukturierten GSN-Diagramms zu unterstützen.

## 6. Zusammenfassung und Ausblick

### 6.2 Ausblick

Das im Sinne dieser Bachelorarbeit entstandene Werkzeug ist dessen Anforderungen gerecht geworden, könnte aber durch verschiedene Änderungen in Zukunft noch verbessert und erweitert werden.

Zum einen könnten die entstandenen Werkzeuge weiterentwickelt werden um auch die Erweiterungen der Goal Structuring Notation zu unterstützen. Beispielsweise die Modularisierung. Dazu müssten verschiedene Darstellungsebenen geschaffen werden. Zum anderen könnten auch sogenannte "Pattern Extensions" umgesetzt werden. Diese würden die Abstraktion der Struktur sowie einzelner Entitäten ermöglichen. Beide hier genannten Erweiterungen könnten auf der jetzigen Implementierung ansetzen und diese erweitern. Es wären keine Änderungen an der bestehenden Architektur erforderlich. Zusätzlich müsste natürlich eine interaktiven Darstellung der Ebenen innerhalb der Modularisierung und einer Darstellung der Kanten der Abstraktionen in den Pattern Extensions implementiert werden.

Außerdem lassen sich die entstandenen Werkzeuge auch an anderen Stellen erweitern. Zum Einen wäre es möglich die Ansicht der eingegebenen Daten innerhalb des Generators verbessern um von der schlichten tabellarischen Lösung auf eine gezeichnete Darstellung umzusteigen. Da gewisse Präferenzen existieren wäre es auch nicht undenkbar die Darstellung innerhalb des Browsers so umzugestalten, dass statt einer tabellarischen Lösung eine gezeichnete Lösung dargestellt wird. Dies hätte natürlich einen dementsprechenden Implementierungsaufwand.

Auch wäre es möglich verschiedene Arten des Exports zu implementieren. Statt nur eine XML-Datei zu generieren könnte die GSN-Struktur auch als Bilddatei gespeichert werden. Beispielsweise in der Formaten .png und .jpeg. Des weiteren besteht auch die Möglichkeit einen PDF-Export zu implementieren.

Eventuell wäre es auch möglich die Analyse und Evaluation eines GSN-Diagrammes zu automatisieren. Eine Schritt-für-Schritt Anleitung um gegebene GSN-Diagramme zu evaluieren. Vor allem für die Struktur und Aufbau des Diagramms sollte es möglich sein gewisse Fehlerquellen auf automatischer Basis zu erkennen.

Die grundlegenden Datenstrukturen innerhalb der Werkzeuge wurden allesamt verständlich und möglichst einfach gehalten um Implementierungen der zuvor beschriebenen Erweiterungen in Zukunft erwägen zu können.

## **Anhang A - Fragebogen der Fall- und Nutzerstudie**

Auf den folgenden Seiten befindet sich der Fragebogen der zur Durchführung der Fall- und Nutzerstudie verwendet wurde.

# Fragebogen zur Goal Structuring Notation Implementierung

Im Rahmen einer Fall- und Nutzerstudie zur Implementierung der Goal Structuring Notation bitte ich Sie, folgenden Fragebogen auszufüllen. Zweck der Studie ist, die Qualität und Nutzbarkeit der entstandenen Tools zu beurteilen.

Laden sie sich hierfür bitte die Goal\_Structuring\_Notation.zip von folgendem Link herunter.  
[https://www.dropbox.com/s/ipp91i2eqr5dbd7/Goal\\_Structuring\\_Notation.zip?dl=0](https://www.dropbox.com/s/ipp91i2eqr5dbd7/Goal_Structuring_Notation.zip?dl=0)

Daraufhin entpacken sie bitte die Goal\_Struturing\_Notation.zip in einem Ordner ihrer Wahl.

**\* Erforderlich**

## Die Darstellung

Der erste Teil der Umfrage bezieht sich auf die Darstellung des GSN-Diagramms innerhalb eines Browsers.

Dazu navigieren Sie in die Ordner GSN\_Example\_1 und öffnen die GSN.xml innerhalb eines Browsers ihrer Wahl, vorzugsweise Mozilla Firefox oder dem Internetexplorer. In Google Chrome lässt sich die Datei nur mit einem Workaround öffnen, dieser befindet sich ebenfalls in der .zip Datei

Der einfachste Weg die GSN.xml Dateien zu öffnen, sind sie in den bereits geöffneten Browser zu ziehen, per drag & drop.

Um die GSN.xml in einem Browser darstellen zu können muss sich die generic.xsl im selben Verzeichnis befinden.

Nun vergleichen Sie bitte die in dem Ordner liegenden Screenshots sowie deren GSN.xml.

1. **Ist die Darstellung sehr viel unübersichtlicher als die, eines gezeichneten GSN-Diagramms? \***

*Markieren Sie nur ein Oval.*

	1	2	3	4	5	
Trifft nicht zu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Trifft zu

2. **Sind die einzelnen Komponenten bzw. GSN-Elemente gut von einander unterscheidbar? \***

*Markieren Sie nur ein Oval.*

	1	2	3	4	5	
Trifft nicht zu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Trifft zu

3. Wenn sie etwas an der Struktur des im Browser dargestellten GSN-Diagrammes ändern könnten, was wäre dies?

.....

.....

.....

.....

.....

4. Wie gefällt ihnen das Farbschema der Darstellung? \*

Markieren Sie nur ein Oval.

	1	2	3	4	5	
Sehr Aufdringlich	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sehr Übersichtlich

5. Soll die Darstellung in einem dezenteren Farbschema bereitgestellt werden?

Markieren Sie nur ein Oval.

☐ Ja

☐ Nein

6. Wenn sie etwas an der Darstellung ändern könnten, was wäre ihre erste Änderung?

.....

.....

.....

.....

.....

7. Kann man sich an die schlichtere tabellarische Darstellung im Browser gewöhnen? \*

Markieren Sie nur ein Oval.

	1	2	3	4	5	
Trifft nicht zu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Trifft zu

## Die XML-Datei

Öffnen sie bitte die beiliegende GSN.xml Datei in einem geeigneten Editor. Ein geeigneter Editor ist beispielsweise Notepad++.

Optional befindet sich auch ein Screenshot Namens XML\_Struktur\_Beiispiel.jpg innerhalb des Ordners.

**8. Ist die Struktur der XML-Datei einleuchtend? \****Markieren Sie nur ein Oval.*

	1	2	3	4	5	
Nein	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Ja

**9. Wie hoch schätzen sie die Einarbeitungszeit die benötigt wird um eine XML dieses Formats selbst zu schreiben?***Markieren Sie nur ein Oval.*

	1	2	3	4	5	
sehr gering	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	sehr hoch

**10. Wie könnte die Struktur der XML Datei noch verbessert werden?**

.....

.....

.....

.....

.....

**11. Ist die Erstellung einer XML-Datei Zeit sparender als die Erstellung eines gezeichnetes GSN-Diagramms?***Markieren Sie nur ein Oval.*

	1	2	3	4	5	
Trifft nicht zu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Trifft zu

**Der GSN-Generator**

Im dritten Teil der Umfrage geht es um den GSN-Generator, einem Tool das für die automatisierte Erstellung von XML Dateien im passenden Schema erstellt wurde.

Dazu öffnen Sie bitte die beigelegte GSN\_Generator.jar Datei, daraufhin sollte sich das Programm öffnen. Falls dem nicht so ist, bitte ich Sie die neueste Version von Java herunterzuladen. Zu finden ist diese auf folgender Seite: <https://www.java.com/de/download/>. Falls das Problem weiterhin bestehen sollte wenden sie sich bitte an mich persönlich, an [plux89@gmail.com](mailto:plux89@gmail.com).

**12. Sieht man dem GSN-Generator seine Funktion an?***Markieren Sie nur ein Oval.*

	1	2	3	4	5	
Trifft zu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Trifft nicht zu

**13. Wie gefällt ihnen das Layout des GSN-Generators?**

Das Layout bezieht sich nur auf die Benutzeroberfläche des Generators  
 Markieren Sie nur ein Oval.

	1	2	3	4	5	
Sehr schlecht	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sehr gut

**14. Wenn sie am Layout des GSN-Generators etwas ändern könnten, was wäre ihre erste Änderung?**

.....

.....

.....

.....

**15. Bewerten Sie Bitte die Handlichkeit des GSN-Generators.**

Markieren Sie nur ein Oval.

	1	2	3	4	5	
Nicht intuitiv Benutzbar	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Intuitiv Benutzbar

**16. Wie wichtig sind ihnen Hotkeys/Shortcuts zur schnelleren Bedienung des GSN-Generators?**

Markieren Sie nur ein Oval.

	1	2	3	4	5	
Unwichtig	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sehr wichtig

**17. Schätzen sie den Zeitaufwand der Erstellung einer XML-Datei, mit dem Generator, geringer ein als ohne diesen?**

Markieren Sie nur ein Oval.

	1	2	3	4	5	
Trifft nicht zu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Trifft zu

**18. Glauben sie es ist möglich mit dem Generator und der Darstellung ein GSN-Diagramm mit weniger Zeitaufwand erstellen zu können, als eines zu Zeichnen?**

Markieren Sie nur ein Oval.

	1	2	3	4	5	
Trifft nicht zu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Trifft zu

Bereitgestellt von



## **Anhang B - Zusammenfassung der Auswertung des Fragebogens**

Auf den folgenden Seiten befindet sich die Zusammenfassung der gegebenen Antworten des Fragebogens.

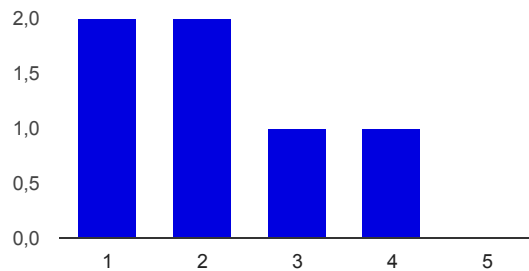
# 6 Antworten

[Alle Antworten ansehen](#)[Analytics veröffentlichen](#)

## Zusammenfassung

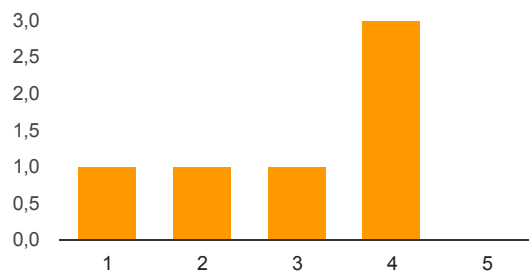
### Die Darstellung

Ist die Darstellung sehr viel unübersichtlicher als die, eines gezeichneten GSN-Diagramms?



Trifft nicht zu:	1	2	33.3 %
	2	2	33.3 %
	3	1	16.7 %
	4	1	16.7 %
Trifft zu:	5	0	0 %

Sind die einzelnen Komponenten bzw. GSN-Elemente gut von einander unterscheidbar?



Trifft nicht zu:	1	1	16.7 %
	2	1	16.7 %
	3	1	16.7 %
	4	3	50 %
Trifft zu:	5	0	0 %

Wenn sie etwas an der Struktur des im Browser dargestellten GSN-Diagrammes ändern könnten, was wäre dies?

Farbliche Abgrenzung der verschiedenen "Ebenen"/Goals, Subgoals

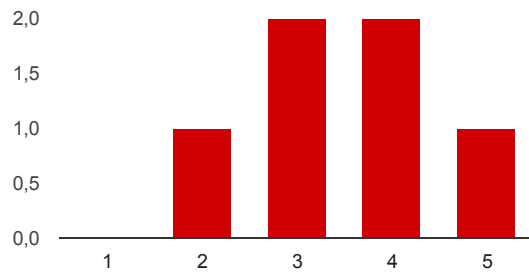
Dickere Ränder um die Elemente?

Wenn ich von der bekannten GS-Notation abweichen müsste, würde ich es am liebsten als eingerückte Textstruktur formatieren. Die Farbgebung kann man als Satzfarbe beibehalten.

Rahmen um alle Elemente wie z.B. Context, Assumption, etc.

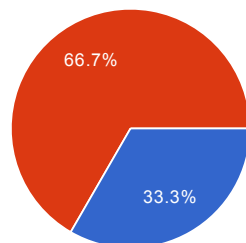
Identische Artefakte nur einmal darstellen.

### Wie gefällt ihnen das Farbschema der Darstellung?



Sehr Aufdringlich: 1	0	0 %
2	1	16.7 %
3	2	33.3 %
4	2	33.3 %
Sehr Übersichtlich: 5	1	16.7 %

### Soll die Darstellung in einem dezenteren Farbschema bereitgestellt werden?



Ja	2	33.3 %
Nein	4	66.7 %

### Wenn sie etwas an der Darstellung ändern könnten, was wäre ihre erste Änderung?

Übersichtlichere, strukturiertere Formatierung.

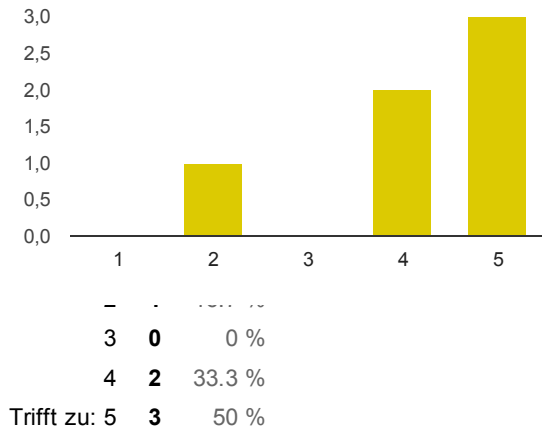
Für jede Ebene eine andere Hintergrundfarbe. Allgemein wäre eine einfachere Unterscheidung der Elemente gut. Das Farbschema ist definitiv nicht zu aufdringlich, es könnte imo noch etwas stärker sein.

Eine Art "Dark Theme", also Grauton wäre super

Einzelne Elemente über Linien verbinden

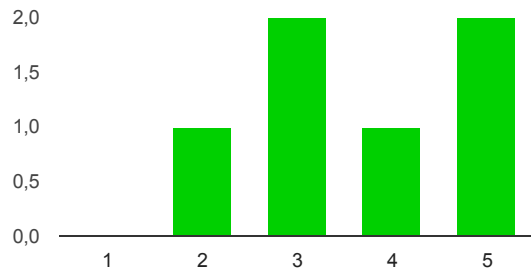
Ggf. durch Numerierung der Elemente die Baumstruktur reflektieren.

### Kann man sich an die schlichtere tabellarische Darstellung im Browser gewöhnen?



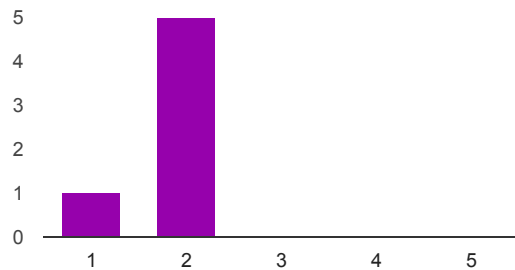
## Die XML-Datei

Ist die Struktur der XML-Datei einleuchtend?



Nein: 1   **0**   0 %  
 2   **1**   16.7 %  
 3   **2**   33.3 %  
 4   **1**   16.7 %  
 Ja: 5   **2**   33.3 %

Wie hoch schätzen sie die Einarbeitungszeit die benötigt wird um eine XML dieses Formats selbst zu schreiben?



sehr gering: 1   **1**   16.7 %  
 2   **5**   83.3 %  
 3   **0**   0 %  
 4   **0**   0 %  
 sehr hoch: 5   **0**   0 %

### Wie könnte die Struktur der XML Datei noch verbessert werden?

Zur Frage Zeichnen vs XML, abhängig von den Kenntnissen des "Programmierers"

Anstatt alle Elemente mit <element> zu taggen, sollte jeder Typ ein eigenes tag haben, z.B. <goal>. Dadurch entfällt auch die Typinformation

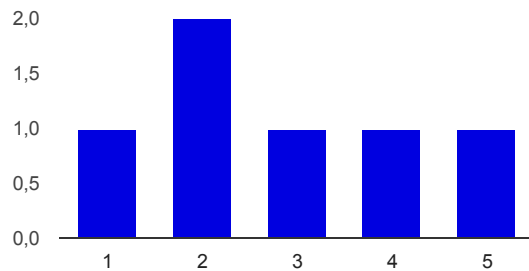
"Anleitung" zur Bedeutung der tags.

keine Meinung

Warum heißt die Wurzel "elements" und die Kinder dieser Wurzel nicht element?

XML ist immer etwas geschwätzig, dafür ist es so minimal wie möglich.

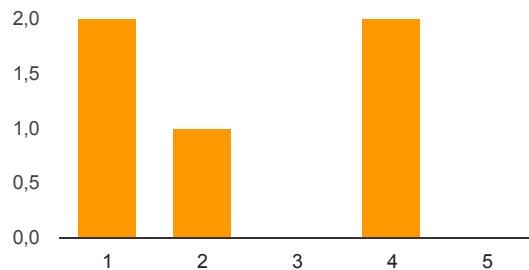
### Ist die Erstellung einer XML-Datei Zeit sparender als die Erstellung eines gezeichnetes GSN-Diagramms?



Trifft nicht zu: 1	1	16.7 %
2	2	33.3 %
3	1	16.7 %
4	1	16.7 %
Trifft zu: 5	1	16.7 %

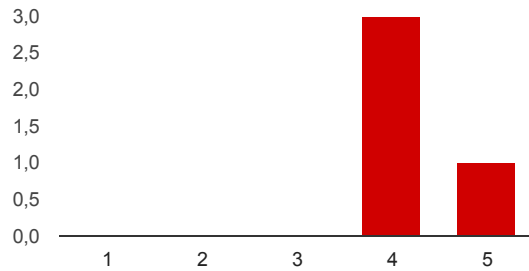
### Der GSN-Generator

#### Sieht man dem GSN-Generator seine Funktion an?



Trifft zu: 1	2	40 %
2	1	20 %
3	0	0 %
4	2	40 %
Trifft nicht zu: 5	0	0 %

### Wie gefällt ihnen das Layout des GSN-Generators?



Sehr schlecht: 1   **0**   0 %  
                   2   **0**   0 %  
                   3   **0**   0 %  
                   4   **3**   75 %  
 Sehr gut: 5   **1**   25 %

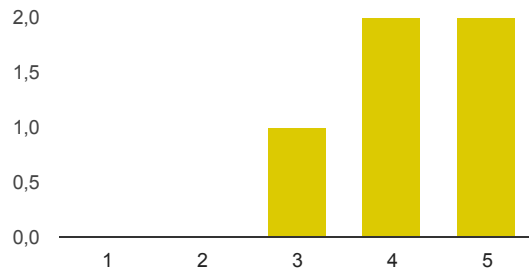
### Wenn sie am Layout des GSN-Generators etwas ändern könnten, was wäre ihre erste Änderung?

java look and feel ausschalten (auf natives OS-look and feel umschalten)

GeneratorGUI Unsupported major.minor version 52.0 Evtl. wäre zur Beantwortung ein png ausreichend gewesen. Installation von SW auf Grund einschränkender Unternehmensregeln nicht einfach möglich.

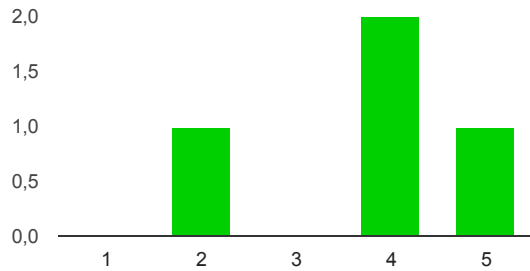
Tooltips hinzufügen.

### Bewerten Sie Bitte die Handlichkeit des GSN-Generators.



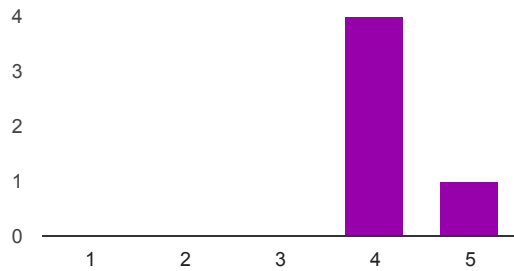
Nicht intuitiv Benutzbar: 1   **0**   0 %  
                                   2   **0**   0 %  
                                   3   **1**   20 %  
                                   4   **2**   40 %  
 Intuitiv Benutzbar: 5   **2**   40 %

### Wie wichtig sind ihnen Hotkeys/Shortcuts zur schnelleren Bedienung des GSN-Generators?



Sehr wichtig: 5    1    25 %

**Schätzen sie den Zeitaufwand der Erstellung einer XML-Datei, mit dem Generator, geringer ein als ohne diesen?**



Trifft nicht zu: 1    0    0 %

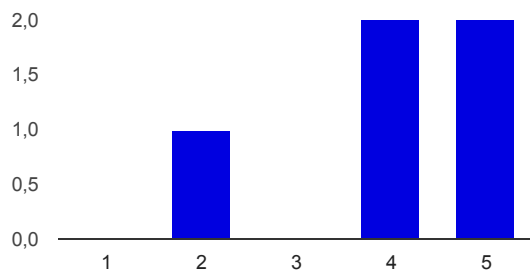
2    0    0 %

3    0    0 %

4    4    80 %

Trifft zu: 5    1    20 %

**Glauben sie es ist möglich mit dem Generator und der Darstellung ein GSN-Diagramm mit weniger Zeitaufwand erstellen zu können, als eines zu Zeichnen?**



Trifft nicht zu: 1    0    0 %

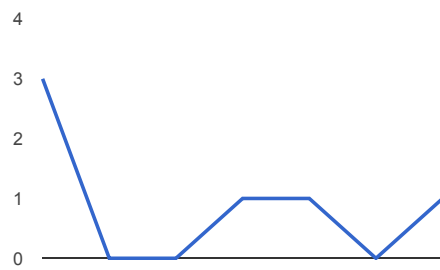
2    1    20 %

3    0    0 %

4    2    40 %

Trifft zu: 5    2    40 %

**Anzahl der täglichen Antworten**



# Abkürzungsverzeichnis

GSN	Goal Structuring Notation
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language
XSLT	XSL Transformation
CSS	Cascading Stylesheets
AADL	Architecture Analysis and Description Language
SIL	Safety Integrity Level
JAXB	Java Architecture for XML Binding
JDK 1.8	Java SE Development Kit 8



# Abbildungsverzeichnis

2.1	Beispiel aus dem GSN Community Standard Version 1 . . . . .	7
2.2	Beispiel eines Ziels aus dem GSN Community Standard Version 1 . . . . .	8
2.3	Beispiel eines Kontext aus dem GSN Community Standard Version 1 . . . . .	9
2.4	Beispiel einer Strategie aus dem GSN Community Standard Version 1 . . . . .	9
2.5	Beispiel einer Annahme aus dem GSN Community Standard Version 1 . . . . .	10
2.6	Beispiel einer Rechtfertigung aus dem GSN Community Standard Version 1 . . . . .	10
2.7	Beispiel einer Lösung aus dem GSN Community Standard Version 1 . . . . .	11
2.8	Beispiel einer UDE aus dem GSN Community Standard Version 1 . . . . .	11
2.9	Beispiel einer Kontext-Von Verknüpfung aus dem GSN Community Standard Version 1 . . . . .	12
2.10	Beispiel einer Unterstützt-Von Verknüpfung aus dem GSN Community Standard Version 1 . . . . .	13
2.11	Eine Skizze der Vorgehensweise in der Sechs-Schritt Methode entnommen aus Kelly [1999a]. . . . .	16
2.12	Eine Skizze der Vorgehensweise in der Bottem-Up Methode entnommen aus Group et al. [2011]. . . . .	18
2.13	Schritte der Evaluation eines GSN-Diagrammes aus dem GSN Community Standard . . . . .	23
4.1	Darstellung der verwendeten Komponentent . . . . .	36
4.3	Beispiel der Struktur einer XML-Datei passenden Schemas . . . . .	37
4.2	Darstellung mit eingebundenem Stylesheet. Beispiel entnommen aus dem GSN Community Standard . . . . .	38
4.4	Layout des GSN-Generators . . . . .	42



# Tabellenverzeichnis

4.1	Tabelle der Kürzel/ID des jeweiligen Typs . . . . .	33
-----	---	----



# Literaturverzeichnis

- [sel 2015] selfhtml.org/wiki. In *Grenzen und Probleme bei der Darstellung mit CSS*, 2015.
- [Bishop et al. 2004] P. Bishop, R. Bloomfield, and S. Guerra. The future of goal-based assurance cases. In *Proc. Workshop on Assurance Cases*, pages 390–395, 2004.
- [Bos et al. 2005] B. Bos, T. Çelik, I. Hickson, and H. W. Lie. Cascading style sheets, level 2 revision 1 css 2.1 specification. *W3C working draft, W3C, June*, 2005.
- [Bray et al. 2011] T. Bray, J. Paoli, C. Sperberg-McQueen, E. Maler, and F. Yergeau. Extensible markup language (xml) 1.0, 2011.
- [Group et al. 2011] G. W. Group et al. Gsn community standard version 1. *Origin Consulting (York) Limited*, 2011.
- [Grunske and Han 2008] L. Grunske and J. Han. A comparative study into architecture-based safety evaluation methodologies using aadl’s error annex and failure propagation models. In *High Assurance Systems Engineering Symposium, 2008. HASE 2008. 11th IEEE*, pages 283–292. IEEE, 2008.
- [Happel et al. 2014] M. Happel, P. Lux, and D. Schwarz. Werkzeuge zur systematischen durchführung einer failure modes effects and diagnostic coverage analysis (fmeda). 2014.
- [Kelly 1999a] T. Kelly. A six-step method for developing arguments in the goal structuring notation (gsn). Technical report, Technical report, York Software Engineering, Flixborough, UK, 1999a.
- [Kelly 2008] T. Kelly. Are safety cases working. *Safety Critical Systems Club Newsletter*, 17 (2):31–33, 2008.
- [Kelly and Weaver 2004] T. Kelly and R. Weaver. *The goal structuring notation—a safety argument notation*. 2004.
- [Kelly 1998] T. P. Kelly. Arguing safety—a systematic approach to safety case management. *Department of Computer Science, The University of York*, 1998.
- [Kelly 1999b] T. P. Kelly. *Arguing safety—a systematic approach to managing safety cases*. University of York, 1999b.
- [Palin et al. 2011] R. Palin, D. Ward, I. Habli, and R. Rivett. Iso 26262 safety cases: compliance and assurance. 2011.

- [Spriggs 2012] J. Spriggs. *GSN-The Goal Structuring Notation: A Structured Approach to Presenting Arguments*. Springer Science & Business Media, 2012.
- [Weaver et al. 2003] R. Weaver, J. Fenn, and T. Kelly. A pragmatic approach to reasoning about the assurance of safety arguments. In *Proceedings of the 8th Australian workshop on Safety critical systems and software-Volume 33*, pages 57–67. Australian Computer Society, Inc., 2003.
- [Werner et al. 2012] P. Werner, S. Gerken, and M. Huhn. Gsn m-edit: Ein modellgetriebener editor für modulare gsn-argumentationen. In *Software Engineering (Workshops)*, pages 163–174. Citeseer, 2012.

### **Erklärung**

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

---

Ort, Datum, Unterschrift