

Software Technik Projekt: CrystalDiskInfo

Computer Engineering, Wintersemester 2019/2020

Philipp Horländer, 566045
Konrad Münch, 565929

Dozent: Prof. Dr. Baar

Berlin, 17. Dezember 2019

Inhaltsverzeichnis

1	Einführung	3
1.1	Aufgabenstellung	3
1.2	Beschreibung der Software	3
2	Modellierung	3
2.1	Use-Case-Modell	3
2.1.1	anzeigenDatenträgerInformation	3
2.1.2	anzeigenDiagramm	4
2.1.3	benachrichtigenAdmin	4
2.1.4	einstellenBenachrichtigung	4
2.1.5	auswählenDatenträger	4
2.2	Domänenmodell	4
2.3	Weitere Modelle	4
3	Geforderte Änderungen	5
4	Modellierung nach den Änderungen	5
4.1	Use-Case-Modell	5
4.1.1	anzeigenDiagramm	5
4.2	Domänenmodell	5
4.3	Weitere Modelle	5
5	Beschreibung der Implementierung	5
6	Fazit	6

1 Einführung

1.1 Aufgabenstellung

Es gilt sich eine Open Source Software auszusuchen und diese zu Reverse-Engineerend abzuändern. Dafür sollen verschiedenen Modelle der Software angefertigt werden, wie Use-Case-, Domänen- und ggf. weitere Modelle. Das ganze soll mit einer Belegarbeit dokumentiert werden.

1.2 Beschreibung der Software

CrystalDiskInfo ist ein nützliches Open-Source Programm für die Bewertung und Überwachung von internen als auch externen HDDs und SSDs. Mit Hilfe der S.M.A.R.T. Technologie (Self-Monitoring, Analysis, Reporting Technology) werden detaillierte Daten, wie zum Beispiel Seriennummer, Temperatur, Lese- und Schreibzyklen, Fehlerrate beim Lesen und Schreiben, Speicher und Puffergrößen, als auch die gesamten Betriebsstunden der Festplatten, aus den Festplatten-Controllern ausgelesen. Diese Daten sind auf dem Controller gespeichert und werden in einem einstellbaren Intervall von der Software ausgelesen und in entsprechenden Dateien auf dem Host-System abgelegt.¹

Wie und welche dieser Daten angezeigt werden lässt sich einstellen. Beispielsweise lässt sich die Seriennummer ausblenden um auf Screenshots nicht aufzutauchen.

Auch lässt sich ein Diagramm über ausgewählte, gespeicherte Daten und Datenträger in Abhängigkeit der Zeit anzeigen.

Eine weitere nützliche Funktion für Administratoren ist es sich bei zu hohen Temperaturen oder einer geringen Lebenszeit der Festplatten benachrichtigen zu lassen. Die geschieht mit einem Warnton, als auch via E-Mail.

Wenn man möchte lassen sich auch Farbschemata der Buttons und Anzeigen ändern.

2 Modellierung

2.1 Use-Case-Modell

[scale=1]Bilder/UseCase1.png

2.1.1 anzeigenDatenträgerInformation

1. Das Programm wird gestartet.
2. Das Programm erfragt die am System angeschlossene Festplatten.
3. Das System gibt die angefragten Daten an das Programm weiter.
4. Das Programm fragt mit SMART die Daten jedes Festplatten Controllers an.

¹Es stellt sich heraus, dass Daten genau dann gespeichert werden, wenn sich etwas am gelesenen Wert ändert. Das heißt, wenn sich zum Beispiel an der Temperatur nach einem Intervall von einer Minute nichts geändert hat, wird auch kein neuer Datenpunkt erstellt.

5. Die Controller geben die Daten an das Programm weiter.
6. Das Programm stellt die Daten lesbar dar.

2.1.2 anzeigenDiagramm

1. Das Programm öffnet ein neues Fenster.
2. Das Programm zeigt einen Graph über eine Standarddatenreihe an.
 - 2.a Der Nutzer wählt eine andere Datenreihe aus.
 - 2.a.1. Das Programm zeigt die andere Datenreihe an.
 - 2.b Der Nutzer wählt Festplatten an/ab.
 - 2.b.1. Das Programm zeigt die gewählten Festplatten-Datenreihen an.

2.1.3 benachrichtigenAdmin

1. Das Programm stellt einen kritischen Wert von Temperatur oder Lebensdauer einer Festplatte fest.
2. Das Programm erstellt eine E-Mail und versendet diese an die eingestellte Adresse.

2.1.4 einstellenBenachrichtigung

1. Das Programm zeigt ein Fenster an mit Feldern für die nötigen Daten.
2. Der Nutzer trägt die nötigen Daten ein und wählt speichern aus.
3. Das Programm speichert die Daten und schließt das Fenster.
 - 2.a Der Nutzer schließt das Fenster.
 - 2.a.1 Das Use-Case endet erfolglos.

2.1.5 auswählenDatenträger

1. Der Nutzer wählt einen anderen Datenträger aus.
2. Das Programm zeigt die Daten des neu gewählten Datenträgers an.

2.2 Domänenmodell

// TODO: Klassendiagramm für die Funktionalität die geändert werden soll.

2.3 Weitere Modelle

// TODO: Welche weiteren Modelle sind sinnvoll?

3 Geforderte Änderungen

Die Funktion zur Anzeige eines Diagramms soll so verändert/erweitert werden, sodass ein "WurmDiagramm (eng. Bubble-Diagram) angezeigt wird. Dabei handelt es sich um eine 2D Darstellung von 3D Datenreihen. Dafür werden zwei Datenreihen auf den Kartesischen Koordinaten dargestellt und eine dritte Datenreihe mit der Größe der Punkte beziehungsweise Blasen. Zum Beispiel auf der x-Achse Temperatur, auf der y-Achse geschriebene Daten und diese Punkte werden nach vergangener Zeit größer.

Die Anforderung wurde dann nach Absprache mit Herrn Prof. Dr. Baar leicht abgeschwächt. Der Grund dafür waren neue Einsichten in das Programm, dessen Aufbau und die Sinnhaftigkeit der Änderung.

4 Modellierung nach den Änderungen

4.1 Use-Case-Modell

Änderungen gab es nur an einem Use-Case.

4.1.1 anzeigenDiagramm

1. Das Programm generiert eine HTML-Datei und öffnet diese im Browser.

4.2 Domänenmodell

4.3 Weitere Modelle

5 Beschreibung der Implementierung

Um das Bubble-Chart darzustellen wurde, wie im Originalprogramm eine JavaScript Bibliothek genutzt. Es wurde sich allerdings für eine andere, modernere Bibliothek entschieden, da die in CrystalDiskInfo verwendete schon sehr alt war und der Anforderungen ein Bubble-Chart darzustellen nicht erfüllen konnte. Deshalb wurde die JavaScript Bibliothek ChartsBundle.js verwendet, welche auch OpenSource auf GitHub.com zur Verfügung steht.

Mithilfe dieser Bibliothek wurde dann eine HTML mit JavaScript erstellt. Um die Verarbeitung in C++ zu vereinfachen wurde das Skript in zwei Teile geteilt und in Text-Dateien abgelegt. Diese Dateien wurden dann in C++ mit den ausgelesenen Daten zu einer fertigen HTML/JS-Datei verknüpft und gespeichert. Die erstellte HTML Datei wird nun über einen MFC² Aufruf im Browser geöffnet.

²MFC ist eine von Microsoft seit 1992 entwickelte Bibliothek zur Erstellung von Windows Desktop Anwendungen.

6 Fazit

Schwierigkeiten traten im gesamten Verlauf zur genüge an. Zuerst war es recht schwer herauszufinden wie CrystalDiskInfo aus dem Source Code funktionsfähig zu bekommen ist. Nötig waren dafür Dateien von der fertigen Installation die dem Source Code nicht beilagen und auch nicht beim Compilieren erstellt wurden.

Als nächstes war es sehr schwer den Code zu verstehen, da es sich bei dem Entwickler von CrystalDiskInfo offensichtlich um einen sehr erfahrenen Programmierer handelt und das Programm entsprechend komplex war. Allerdings ist CrystalDiskInfo sehr spärlich kommentiert. Das machte das verstehen deutlich schwerer.

Die geforderte Änderung umzusetzen erwies sich als schwerer als erwartet.