

Constraint-basierte Planung und Optimierung von Prüfungsterminen mithilfe einer graphischen Benutzeroberfläche

Thomas Kuhnke¹ und Tom Hausten²

Abstract: In zwei Bachelorarbeiten ([Ha13], [Ku13]) wurde die constraint-basierte Prüfungsplanung mittels einer dafür erstellten graphischen Benutzeroberfläche umgesetzt. Dabei sollte mit der graphischen Benutzeroberfläche eine manuelle Planung von Prüfungen sowie das Anstoßen einer automatisierten Planung ermöglicht werden. Die automatisierte Planung sollte hierbei feststehende und von dem Anwender individuell festgelegte Bedingungen berücksichtigen und den Prüfungsplan möglichst nach diesen Kriterien optimieren. Bei der Planung sollten alle an der BTU Cottbus – Senftenberg angebotenen Veranstaltungen und Prüfungsordnungen berücksichtigt werden. Um die Komplexität der Planung zu reduzieren und eine bessere Übersichtlichkeit zu gewährleisten wurde die Planung in die drei Teilbereiche Tages-, Block- und Raumplanung unterteilt.

Keywords: Choco, Constraint-Programmierung, GUI, Prüfungsplanung, Vaadin

1 Einleitung

Planungsprobleme sind Probleme, die jeder Person in unterschiedlicher Ausprägung und Schwierigkeit begegnen können. Sie können im privaten Bereich aber auch im unternehmerischen Bereich auftreten. Planen allgemein ist das Erzeugen einer zeitlich geordneten Folge von Aktionen, die von der Ausgangssituation in die Zielsituation führt. Es gibt verschiedene Arten von Planungen. Die Bachelorarbeiten beschäftigen sich mit dem Scheduling. Scheduling ist ein Entscheidungsprozess zum Verteilen von Aufgaben und Tätigkeiten über einen gewissen Zeitraum mit begrenzten Ressourcen, welcher nach bestimmten Kriterien optimiert wird. Unter Ressourcen können beispielsweise die Besprechungsräume eines Unternehmens verstanden werden, die nur in gewisser Anzahl vorhanden sind. In diesem Fall könnte ein Ziel der Planung sein, maximal so viele Besprechungen gleichzeitig durchzuführen, wie es Besprechungsräume gibt [Pi12].

Bei der Prüfungsplanung an der BTU Cottbus – Senftenberg handelt es sich um ein komplexes Zeitplanungsproblem, welches derzeit in einem langwierigen Prozess manuell von den Mitarbeitern der Universität erstellt wird. Mehrere 100 Prüfungen müssen auf zwei Prüfungszeiträume à zwei Wochen verteilt werden. Dabei wird primär darauf geachtet, dass möglichst kein Student zwei Prüfungen an einem Tag schreiben

¹ BTU Cottbus – Senftenberg, Platz der Deutschen Einheit 1, 03046 Cottbus, thomas.kuhnke@b-tu.de

² BTU Cottbus – Senftenberg, Platz der Deutschen Einheit 1, 03046 Cottbus, tom.hausten@b-tu.de

muss und die Prüfungen möglichst zwischen Montag und Freitag, im Ausnahmefall auch Samstag, stattfinden. Somit stehen 24 Tage für die Prüfungsplanung zur Verfügung. Das Problem der Planung liegt hier in der eingeschränkten Ressourcenverfügbarkeit von Zeit- und Prüfungsräumen, in denen Prüfungen durchgeführt werden können. Zudem müssen bestimmte Bedingungen beachtet werden. Beispielhaft ist dafür zu nennen, dass Professoren bzw. Studenten nicht zwei Prüfungen gleichzeitig haben dürfen. Es gibt keine feste Prüfungsdauer. In der Regel dauert eine Prüfung 90 Minuten, jedoch können diese auch 60 oder 180 Minuten dauern. Das ist abhängig von der Vorgabe der einzelnen Professoren und deren Lehrstuhl. An der BTU Cottbus – Senftenberg gibt es über 50 Prüfungsordnungen und über 1000 Veranstaltungen, die bei der Planung in unterschiedlicher Art und Weise berücksichtigt werden müssen.

Aufgrund der großen Anzahl an Informationen und Abhängigkeiten ist es sehr schwer und zeitaufwändig eine funktionierende Planung zu erstellen. Zwar ist es möglich sich an vorherigen Lösungen zu orientieren, aber trotzdem bleibt der Prozess aufwändig und fehleranfällig. Aus diesem Grund sollte ein automatisiertes Verfahren entwickelt werden, um die Planungen zu vereinfachen. Dazu gehört die automatische Prüfungsplanerstellung auf Basis der constraint-basierten Programmierung durch ein Tool sowie das Anpassen bzw. die manuelle Erstellung eines Prüfungsplans über eine graphische Benutzeroberfläche.

2 Die constraint-basierte Planung und Optimierung von Prüfungsterminen

Eine initiale Variante eines Planungstools hat die Prüfungsplanung als ein gesamtes Problem behandelt und nach einer Gesamtlösung gesucht. Dabei wurden ca. 80 Prüfungen aus vorhandenen CSV-Dateien eingelesen. Es wurden jedoch keine Optimierungen vorgenommen, was zu einer ungleichmäßigen Verteilung führte. Dieses Problem sollte mit dem neuen Planungstool beseitigt werden.

2.1 Umsetzung

Der objekt-orientierte Anteil des Planungstools wurde mit Java umgesetzt. Mithilfe der Constraint-Bibliothek Choco (Version 2.1.5) wurde das Konzept der constraint-basierten Programmierung durchgeführt. Dabei wurde auf einer SQL-Datenbank mit ca. 400 Prüfungen gearbeitet, welche unter anderem Informationen über teilnehmende Studiengänge und zeitliche Einordnungen in den Studiengängen beinhaltet. Um den Aufwand des Constraint-Lösers zu minimieren, wurde das Gesamtplanungsproblem in mehrere Teilprobleme aufgeteilt. Zunächst werden die Prüfungen auf die jeweiligen Tage verteilt, danach werden sie auf Zeitblöcke innerhalb der Tage verteilt und zum Schluss wird nach einer passenden Raumkonstellation gesucht, wie in Abb. 1 graphisch dargestellt wird. Dabei werden die Zwischenergebnisse gespeichert, so dass Änderungen

an den Block- bzw. Raumplanungsconstraints keine erneute Planung der vorangehenden Phase verursacht.

2.2 Ergebnis

Im Allgemeinen eignet sich die Constraint-Programmierung sehr gut für Planungsprobleme. Es hat sich gezeigt, dass die Optimierungsmethoden, welche Choco anbietet, durchaus vorteilhaft für die Planung sein können. Mithilfe der Optimierung ist es möglich die Nichteinhaltung von Constraints auf ein Minimum zu reduzieren. Allerdings hat das Planungstool auch gezeigt, dass Planungen schnell zu komplex werden können für den Constraint-Löser, was eine stark erhöhte Laufzeit zur Folge haben kann. Bei den 400 Prüfungen der BTU müssen allein bei der Tagesplanung aufgrund der paarweisen Überprüfung mehr als 1000 Constraints betrachtet werden. Die Komplexität der Planung ist daher sehr stark abhängig von der Menge der Prüfungen, Studiengängen sowie dem Prüfungszeitraum. Im Fall der Prüfungsplanung der BTU Cottbus – Senftenberg haben die Ergebnisse auch gezeigt, dass die vier Wochen Prüfungszeitraum für alle Eventualitäten nicht ausreichend sind und somit Kompromisse eingegangen werden müssen. Ein direkter Vergleich mit anderen Constraint-Lösern wie Gecode (Constraint-Bibliothek für C++) konnte im Rahmen der Bachelorarbeit nicht durchgeführt werden, da dieser eine fast vollständige Re-Implementierung zur Folge gehabt hätte.

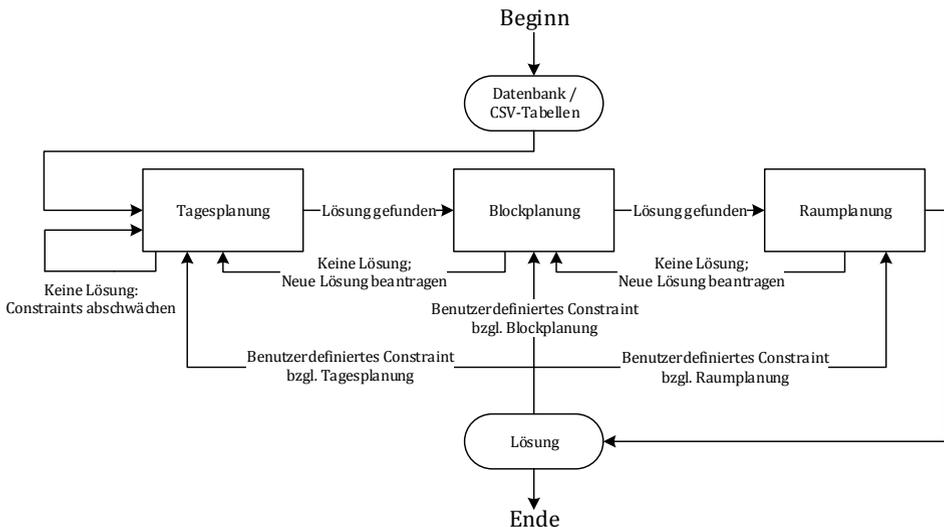


Abb. 1: Konzept zum Ablauf des Planungstools

3 Die graphische Benutzeroberfläche zur Planung und Verwaltung von Prüfungsterminen

Die graphische Benutzeroberfläche stellt einen wichtigen Teil bei der Planung von Prüfungen dar. Sie bietet dem Nutzer die Möglichkeit des manuellen Planens aller Prüfungen in den Prüfungszeiträumen und ist gleichzeitig eine Schnittstelle zum constraint-basierten Tool für die automatisierte Planung und Optimierung der Prüfungen.

3.1 Anforderungen

Die graphische Benutzeroberfläche soll zur Vereinfachungen und als Unterstützung für das Planen von Prüfungsterminen dienen. Sie soll die Möglichkeit bieten, prüfungsrelevante Daten zu bearbeiten und neue Daten hinzuzufügen. Alle relevanten Daten müssen übersichtlich und möglichst einfach dargestellt werden. Zudem sollte das Planen von Prüfungen in den drei Teilplanungen möglich sein. Jede einzelne dieser Ebenen ist frei planbar, wobei die Prüfungsplanung immer mit der Tagesplanung beginnt und mit der Raumplanung endet. Es sollte eine Schnittstelle zum constraint-basierten Planungstool geben, um Prüfungen automatisiert zu planen. Die Ergebnisse des Constraint-Lösers müssen in der Benutzeroberfläche darstellbar und zudem auch durch den Nutzer anpassbar sein. Dem Benutzer soll zudem auch die Möglichkeit gegeben sein, für einzelne Prüfungen zusätzliche Bedingungen anzugeben, welche von dem constraint-basierten Tool berücksichtigt werden. Dies kann beispielsweise ein festgelegter Termin einer Prüfung sein. Somit hat die graphische Oberfläche zur Aufgabe, das komplexe Problem möglichst einfach und übersichtlich darzustellen.

3.2 Umsetzung

Um langwierige Installationen eines Tools bei den Nutzern zu verhindern, wurde eine webbasierte Lösung angestrebt. Die Oberfläche wurde mit Framework Vaadin umgesetzt. Vaadin ist ein auf AJAX basierendes Web Application Framework, welches Ähnlichkeiten zu GWT aufweist. Anwendungen hierfür lassen sich in Java programmieren und können mit HTML und CSS verfeinert werden. Dabei läuft die komplette Logik der Anwendung auf einem Server und kann über einen Browser von den Benutzern genutzt werden. Dadurch ist das Tool plattform- und ortsunabhängig und bietet eine hohe Kompatibilität.

Alle Daten, welche für die Prüfungsplanung benötigt werden, sind in einer Datenbank hinterlegt. Die Benutzeroberfläche bietet eine tabellarische Darstellung aller zur Prüfungsplanung benötigten Daten aus der Datenbank an, in welcher die Daten bearbeitet und hinzugefügt werden können. Auch das automatische Planen der Prüfungen aus den vorbereiteten Daten aus der Datenbank wird unterstützt. In der Abb. 2 ist der Ablauf einer automatisierten Planung durch das constraint-basierte Tool aus der graphischen Benutzeroberfläche heraus dargestellt. Die fünf dargestellten Schritte

können für alle drei Teilplanungen Tages-, Block- und Raumplanung durchgeführt werden. Die Lösungen aller Planungsebenen können, wie in Abb. 3 dargestellt, in einer kalenderähnlichen Darstellung angeschaut und manuell angepasst werden. Somit bietet die Benutzeroberfläche einen hohen Planungskomfort. Die Oberfläche bietet auch die Möglichkeit, mehrere Prüfungsplanungen für einen Zeitraum zu erstellen. Zudem werden gleichzeitige Planungen mehrerer User unterstützt.

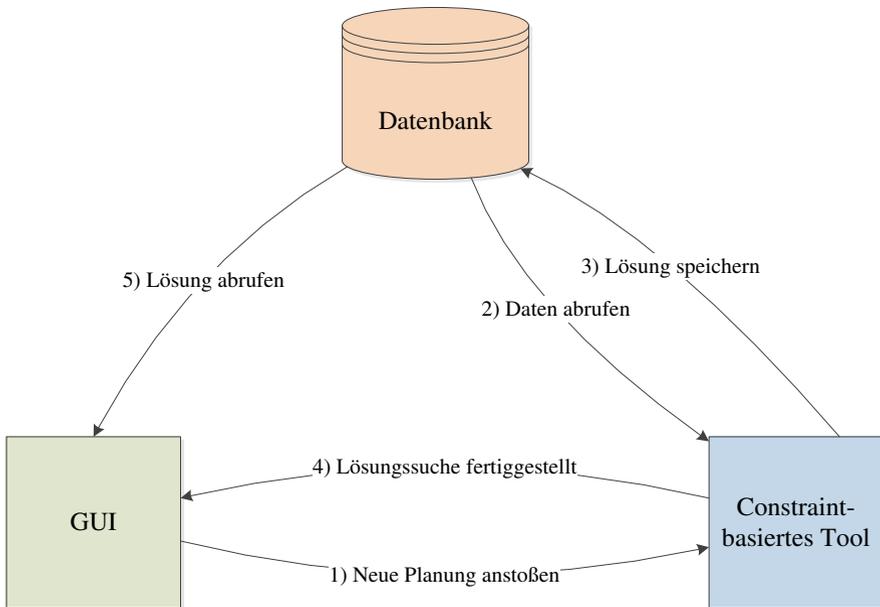


Abb. 2: Ablauf der automatisierten Planung mithilfe des constraint-basierten Tools



Abb. 3: Ausschnitt der Blockplanung

Literaturverzeichnis

- [Ha13] Hausten, T.: Entwurf und Implementierung einer graphischen Benutzeroberfläche zur Planung von Prüfungsterminen. BTU Cottbus – Senftenberg, unveröffentlichte Bachelorarbeit, 2013.
- [Ku13] Kuhnke, T.: Constraint-basierte Planung und Optimierung von Prüfungsterminen. BTU Cottbus – Senftenberg, unveröffentlichte Bachelorarbeit, 2013.
- [Pi12] Pinedo, M.: Scheduling: Theory, Algorithms and Systems. Springer US, 2012.