



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Seminararbeit AW1

André Goldflam

webbasiertes kollaboratives
Konfigurationssystem

Inhaltsverzeichnis

1	Einführung	3
1.1	Aufbau der Arbeit	3
2	Problemstellung und Grundlagen	4
2.1	Problemstellung (Szenario)	4
2.2	Abstraktion der Problemstellung	5
2.3	Knowledge Engineering	6
2.3.1	Konstruktionssystem	6
2.3.2	Methoden zur Konfigurierung	7
2.4	Constraints	9
2.4.1	Lösungsverfahren	10
2.4.2	Fazit	11
2.5	CSCW - Computer Supported Collaborative Work	12
2.5.1	Klassifizierung von CSCW-Systemen	12
2.5.2	3K-Modell	12
2.5.3	Fazit	13
3	Aktuelle Produkte und Arbeiten	15
3.1	Kollaborations-Systeme	15
3.1.1	CommSy	15
3.2	Constraint-Systeme	16
3.2.1	JCL	16
3.2.2	ILog JSolver	17
3.2.3	JESS	17
3.2.4	Fazit	17
3.3	verwandte Arbeiten	18
4	Ausblick und Risiken	19
4.1	Ausblick	19
4.2	Risiken	20
	Literaturverzeichnis	21

1 Einführung

Der Mensch ist mit immer komplexer werdenden Sachverhalten konfrontiert. Nicht selten müssen Formulare ausgefüllt werden, bei denen das eigene Fachwissen nicht unbedingt ausreichend ist. Zusätzlich werden aus Kostengründen viele bürokratischen Vorgänge mittlerweile online abgewickelt, z.B. ist es anhand der ELSTER-Software¹ möglich, die eigene Steuererklärung über Internet an das Finanzamt zu übermitteln. Dabei wächst die Verantwortung der Nutzer solcher Software, ohne dass ihnen entsprechende zusätzliche Hilfestellungen gegeben werden. In dieser Ausarbeitung sollen die Grundlagen für ein System vorgestellt werden, welches Nutzer bei der Ausarbeitung eines komplexen Formulars unterstützt und ihnen zusätzlich adäquate Unterstützungsmaßnahmen bietet.

1.1 Aufbau der Arbeit

Im folgenden Kapitel soll ein Szenario vorgestellt werden und anschließend auf die wissenschaftlichen Teilbereiche abstrahiert werden. Diese lauten „Knowledge Engineering“, „Constraints“ und „Computer Supported Collaborative Work“. Innerhalb der Abschnitte soll auf die für die Problemstellung relevanten Thematiken eingegangen und am Ende in einem Fazit zusammengefasst werden. Anschließend sollen aktuelle Produkte aus diesem Bereich dargestellt und deren Nutzen in Bezug auf die Problemstellung erläutert werden. Zusätzlich sollen verwandte Arbeiten welche an der HAW Hamburg erstellt worden sind kurz vorgestellt werden. Abschließend soll ein Ausblick auf die zu erwartenden Schritte im Rahmen des AW2-Seminars gegeben und auf mögliche Risiken eingegangen werden.

¹<https://www.elster.de/>

2 Problemstellung und Grundlagen

2.1 Problemstellung (Szenario)

Ein Arbeitnehmer (im folgenden Benutzer genannt) soll seine Steuererklärung erstellen. Hierzu soll ihm ein System zur Verfügung gestellt werden, welches ihn bei der Eingabe der Daten unterstützt. Ein fehlerhaftes Ausfüllen des Formulars auf Grund eines Missverständnisses soll dabei vermieden werden, da dies negative Auswirkung auf die zu zahlenden Steuern haben kann.

Die Unterstützung soll dabei auf folgende Arten gewährleistet sein: Zum einen kann der Benutzer selbst das Formular ausfüllen, wobei die Unterstützung in Form von Musterbeispielen oder zusätzliche Informationen genutzt werden kann. Zusätzlich soll das Formular nur die für den Benutzer notwendigen Fragen präsentieren, da etwaige nicht relevante Informationen zu einer Irreführung des Benutzers führen können. Weiterhin unterliegt ein Steuerformular gewissen Regeln, z.B. müssen einige Bereiche nur ausgefüllt werden wenn an einer bestimmten Stelle im Formular eine Angabe gemacht wurde oder der Inhalt eines bestimmtes Feldes ist von den Inhalten mehrere Felder abhängig. Dabei muss der Inhalt eines Feldes nicht unbedingt einen bestimmten Wert annehmen sondern kann auch aus einem von den vorherigen Eingaben abhängigen Wertebereich stammen.

Zum anderen soll der Benutzer die Möglichkeit haben einen Steuerberater (im folgenden Berater genannt) in den Prozess mit einzubeziehen, um mögliche Fragen zu klären oder um eine Endkontrolle durchzuführen. Der Benutzer sammelt dabei seine Fragen und kommuniziert diese dem Berater. Der Berater kann auf das ausgefüllte Formular zugreifen und die Fragen entweder textuell beantworten oder selbst direkt lösen, indem er die Änderungen an dem Formular für den Benutzer vornimmt. Nun kann der Benutzer anhand der beantworteten Fragen das Formular weiter ausfüllen und wiederum wenn nötig weitere Fragen stellen. Am Ende dieses Prozesses soll der Berater eine Endkontrolle durchführen.

Die Kommunikation soll dabei direkt zwischen Benutzer und Berater in Form einer Webkonferenz gestaltet werden. Diese Konferenz soll nahtlos in den Erstellungsprozess integriert werden indem der Steuerberater genau wie der Kunde das entsprechende Formular zur gleichen Zeit bearbeiten können soll.

2.2 Abstraktion der Problemstellung

In diesem Abschnitt soll die Problemstellung abstrahiert und der Bezug zu Forschungsgebieten der Informatik hergestellt werden. Folgend werden die Bereiche kurz eingeführt und in einem anschließenden Abschnitt weiter beschrieben.

Die vorgestellte Anwendung unterstützt den Benutzer auf Basis des implementierten Wissens der Steuererfassung hinsichtlich einer korrekten Bearbeitung des Formulars. Ein System dieser Art fällt in den Bereich des Knowledge Engineering aus dem Forschungsgebiet der künstlichen Intelligenz.

Weiterhin besteht das Formular nicht nur aus den Feldern und den zugehörigen Texten sondern auch aus Musterbeispielen, Hilfsinformationen, Regeln welche die Zusammenhänge zwischen den Feldern beschreiben, den Fragen des Benutzers und eventuell den Antworten des Beraters. Das Formular enthält zusätzliche Informationen welche Abhängigkeiten innerhalb des Formulars darstellen und kann als ein komplexes Dokument bezeichnet werden. Die bisher als „Regeln“ beschriebene Abhängigkeiten werden im Bereich der künstlichen Intelligenz als so genannte Constraints modelliert.

Der Benutzer und der Berater arbeiten gemeinsam auf die Erstellung der Steuererklärung hin. Dabei sollen sie wie beschrieben anhand einer Webkonferenz und einem System unterstützt werden, welches das gleichzeitige Bearbeiten des Steuerformulars ermöglicht. Das gemeinsame, oder auch kollaborative Arbeiten stammt aus dem Bereich des Computer Supported Collaborative Work.

2.3 Knowledge Engineering

Der Begriff wurde 1983 von E. Feigenbaum bezüglich der sog. Expertensysteme eingeführt und umfasst alle Tätigkeiten, Überlegungen zur Erfassung, Verwaltung, Verwendung und Transformation von Wissen (Vgl. [Görz u. a. \(2003\)](#), S.599). Interessant für diese Ausarbeitung ist das Thema der Problemlösungsmethoden. Problemlösungsmethoden bestimmen wie das gegebene Wissen zur Lösung des Problems genutzt wird. Dabei wird grundsätzlich zwischen zwei unterschiedlichen Anwendungsbereichen unterschieden: Dem analytischen und dem synthetischen Anwendungsbereich. Diese sind wie folgt charakterisiert:

analytisch Es wird eine Problemlösung aufgrund einer Klassifikation des Problems ausgewählt (Klassifikation, Diagnostik).

synthetisch Die Problemlösung wird aus einzelnen primitiveren Bausteinen konstruiert (Konstruktion).

Die Problemklasse der Konstruktion untergliedert sich weiter in die Teilbereiche wie z.B. Konfiguration, Planung, Scheduling.

Besonders interessant im Rahmen dieser Arbeit ist der Teilbereich der Konfiguration, die anderen bereits genannten Bereiche werden hingegen vernachlässigt. (Vgl. [Görz u. a. \(2003\)](#), S.617)

2.3.1 Konstruktionssystem

Eine Konfigurationsaufgabe kennzeichnet sich durch die Art und Weise wie die Lösung gefunden wird. Diese ergibt sich aus einer Reihe von Konfigurationsschritten welche aus Einzelkomponenten bestehen (vgl. [Cunis und Günter \(1991\)](#) S.37 [zitiert nach [Runte \(2006\)](#) S. 10]). Nach (vgl. [Günter \(1991\)](#) S.2 ; vgl. [Brauer:1987 \(1987\)](#) S. 349 [zitiert nach [Runte \(2006\)](#) S. 10]) weisen Aufgaben aus dem Bereich der Konfiguration folgende Charakteristika auf:

Großer Lösungsraum Die Menge der möglichen Konfigurationskombinationen kann sehr groß sein.

Rücknahme von Entscheidungen Es müssen eventuell bestimmte Entscheidungen im Verlauf der Konfiguration zurückgenommen werden, um zu einer gewünschten Lösung zu gelangen.

Hierarchisches Vorgehen In vielen Anwendungsbereichen ist die Konfiguration eine Hierarchie von Komponenten. Dementsprechend ist für den Ablauf der Konfiguration ebenso eine hierarchische Gliederung zu empfehlen.

Behandlung von Abhängigkeiten Zwischen den Einzelkomponenten aus denen die Lösung aufgebaut ist existieren bestimmte Abhängigkeiten, welche die Kombinationsmöglichkeiten der Komponenten beschreiben. Daher sind Repräsentation und Verarbeitung dieser Abhängigkeiten ein wichtiger Bereich innerhalb der Konfigurationsdomäne.

2.3.2 Methoden zur Konfigurierung

Nach (Runte (2006) S. 11) legen Konfigurationsmethoden zum einen die Art der Wissensmodellierung und zum anderen die die Weise fest, wie eine Konfigurationsentscheidung getroffen wird. Für die vorliegende Problemstellung sind insbesondere die zwei folgenden Methoden interessant.

Fallbasiertes Konfigurieren

Nach (vgl. Runte (2006) S. 15) existiert bei dieser Methode als Grundlage zur Konfiguration eine sogenannte Fallbibliothek, welche aus dem Wissen von gelösten Konfigurationsproblemen besteht. Nun sollen ähnliche Konfigurationsprobleme unter der Annahme bewältigt werden, dass ähnliche Probleme zu ähnlichen Lösungen führen. Es können zwei grundsätzlich verschiedene Vorgehensweisen angewendet werden (Günther und Kühn (1999) S. 53 [zitiert nach Runte (2006) S. 15]):

transformation analogy Einen ähnlichen Fall suchen und diesen hinsichtlich der Lösung des eigenen Problems abändern

derivational analogy Für jeden einzelnen Schritt das fallbasierte Wissen anwenden. Die Fallbibliothek kann als Kontrollwissen genutzt werden.

Bei beiden Varianten muss das ausgewählte fallbasierte Wissen für das aktuelle Problem angepasst werden. In der Regel sind die Anpassungen zudem recht umfangreich. Ein weiterer Nachteil dieser Methodik besteht zudem in der Tatsache, dass fallbasierte Systeme verallgemeinernde Lösungen bevorzugen. Weiterhin gibt es bei der Nutzung dieser Methodik keine kausale Erklärung für das Zustandekommen einer bestimmten Lösung (vgl. Runte (2006) S. 16).

Constraint-basiertes Konfigurieren

Ein Constraint (ins Deutsche übersetzt „Bedingung“) legt eine Beziehung zwischen den Konfigurationsvariablen fest. Auf diese Weise können Abhängigkeiten zwischen Objekten repräsentiert und ausgewertet werden. Anhand des Wissens über das spezielle Anwendungsgebiet können die Constraints definiert werden. In Abschnitt 2.4 wird näher auf Eigenschaften sowie Lösungsverfahren eingegangen. Die Konfiguration stellt zusätzliche Anforderungen an ein Constraint-System:

- Das Constraint-Netz wird während des Konfigurierungsvorgangs inkrementell aufgebaut. Dies bedeutet, dass die vollständige Anzahl von Constraints erst bekannt ist, wenn die Lösung vorliegt. Ein Constraint-System sollte entsprechend einen inkrementellen Aufbau des Constraint-Netzes unterstützen.
- Je nach Anwendungsgebiet sollte das Constraint-System bestimmte Constraint-Typen unterstützen (z.B. räumliche, zeitliche oder numerische)
- die Komplexität von Constraint-Problemen aus dem Bereich NP-vollständig erlaubt meistens nicht die Propagation jedes einzelnen Wertes sondern nur von dessen Ober- und Untergrenzen. Daher sollte das Constraint-System einen Kompromiss zwischen Effizienz und Qualität anbieten. Abhängig vom Anwendungsfall sollte es entsprechend möglich sein, ein bestimmtes Verfahren zur Lösung von Constraints mit der nötigen Korrektheit nutzen zu können.

Ein Constraint-System welches alle genannten Anforderungen erfüllt ist nach (Runte (2006) S.14) derzeit nicht bekannt. Daher sind Konfigurationswerkzeuge entweder auf externe Komponenten oder eigene Implementierungen angewiesen.

Fazit

Die Methodik des fallbasierten Konfigurierens stellt sich für die in dieser Ausarbeitung vorgestellte Problemstellung als unbrauchbar dar. Dies ist darauf zurückzuführen, dass die Erstellung einer Steuererklärung individueller Natur ist und Unterschiede zwischen den einzelnen Fällen nicht verallgemeinerbar sind. Eine durch die Fallbibliothek vorgeschlagene Musterlösung zu adaptieren kann den Benutzer leicht verwirren und irreführen. Über Constraints können sich die Abhängigkeiten zwischen verschiedenen Kriterien des Steuerrechts flexibel darstellen lassen. Die Constraints dienen in diesem Falle als Kontrollstruktur für die Eingaben des Benutzers und gewährleisten die formale Korrektheit der Eingaben.

2.4 Constraints

Laut (Hentenryck und Saraswat (1996), S.701) können Constraints als Restriktionen angesehen werden, welche auf eine Menge von Möglichkeiten angewendet werden kann. Mathematische Constraints gelten als genau definierte Beziehungen zwischen Variablen. Ein Constraint schränkt dabei die mögliche Werte ein, die eine Variable annehmen kann und enthält durch diese einschränkende Beziehung eine Information über die Variable.

Weiterhin besitzen Constraints nach (Hentenryck und Saraswat (1996), S.701 - 702) folgende Eigenschaften:

1. Wie bereits oben angemerkt beschreiben Constraints eine Beziehung zwischen Variablen
2. Die Ausführungsreihenfolge von Constraints wirkt sich nicht auf die Lösung aus. Z.B. kann für $c1 \ X + Y \geq Z$ und für $c2 \ X + Y \leq z$ festgelegt werden.
3. Constraints sind untereinander selten unabhängig. Z. B. folgt aus $c1$ und $c2$, dass $X + Y = Z$ lauten ohne, dass dies explizit angegeben werden muss
4. Constraints sind nichtdirektional.
5. Constraints beschreiben Beziehungen zwischen Variablen, ohne zu definieren auf welche prozedurale Arte die Beziehung festgestellt wird

Anschließend werden wichtige Begriffe aus dem Bereich der Constraints erläutert.

Unäre und binäre Constraints Ein Constraint bestehend nur aus einer (unär) oder zwei (binär) beteiligten Variablen (Vgl. Görz u. a. (2003) S. 268).

Constraint-Netz Stehen mehrere Constraints miteinander in Beziehung ergibt sich daraus ein sogenanntes Constraint-Netz. Abbildung 2.1 zeigt exemplarisch ein solches Netz bestehend aus den Constraints $c1 = x < y$, $c2 = y < z$ und $c3 = x < z$. Man spricht von einem binären Constraint-Netz, wenn es ausschließlich aus unären und binären Constraints besteht. (Vgl. Görz u. a. (2003) S. 268)

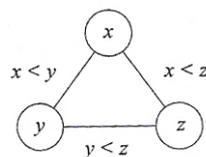


Abbildung 2.1: Beispielhaftes Constraint-Netz, Quelle: Görz u. a. (2003) S.269

Constraint-Erfüllung Eine bestimmte Wertezuweisung erfüllt einen Constraint wenn dessen Bedingung nicht verletzt. Bezogen auf das vorhergehende Beispiel in Abbildung 2.1 würde die Wertezuweisung $(x,y,z) = (4,6,6)$ zwar die Constraints c_1 und c_3 , nicht aber $c_2 = y < z$ erfüllen. Erfüllt eine Wertezuweisung alle Constraints des Netzes ist sie eine Lösung. (Vgl. Görz u. a. (2003) S. 268)

Lösung Ein Constraint-Netz gilt als konsistent wenn es eine Lösung besitzt, andernfalls wird es als inkonsistent bezeichnet. Die Lösungsfindung eines Constraint-Netzes wird im allgemeinen als Constraint-Erfüllungsproblem (Constraint-Satisfaction-Problem) bezeichnet. Nach (vgl. Görz u. a. (2003) S. 269) sind Constraint-Erfüllungsprobleme in der Regel NP-vollständig. Die Oberklasse NP bezeichnet eine bestimmte Kategorie von Problemen, die auf nicht-deterministischen Maschinen in polynomieller Zeit gelöst werden können. Die schwierigsten Probleme innerhalb dieser Klasse werden als NP-vollständig bezeichnet. Sie gelten als nicht-handhabbar, da der Zeitaufwand zur Lösung eines Problems dieser Klasse extrem hoch ist (vgl. Görz u. a. (2003) S.183). An dieser Stelle soll es reichen, dass die Lösung eines solchen Problems nicht trivial ist und daher spezielle Verfahren angewendet werden auf die nachfolgend eingegangen wird. Eine nähere Betrachtung der Komplexitätstheorie wird in Garey und Johnson (1979) durchgeführt.

Konsistenz Viele Algorithmen zur Lösungsfindung erzeugen einen bestimmten Grad lokaler Konsistenz innerhalb des Constraint-Netzes, bevor die eigentliche Lösung berechnet wird. Durch die Erzeugung der lokalen Konsistenz können bestimmte Wertebereiche ausgeschlossen werden, die nicht zur Lösung des Netzes beitragen können. (Vgl. Görz u. a. (2003) S. 269 - S. 271)

2.4.1 Lösungsverfahren

Laut (Hentenryck und Saraswat (1996) S. 707) können die Verfahren um Constraint-Erfüllungsprobleme zu lösen in zwei verschiedene Verfahrensarten unterteilt werden: Systematische Verfahren, welche mit großem Zeitaufwand den gesamten Lösungsraum durchlaufen und die optimalste Lösung finden und stochastische Verfahren, welche mit vergleichsweise geringem Zeitaufwand eine korrekte aber nicht unbedingt optimalste Lösung finden. (Hentenryck und Saraswat (1996) S. 707) nennt im Bereich der systematischen Suche den Backtracking Algorithmus und Erweiterungen dessen. Im Bereich der stochastischen Suche wird die gierige Suche genant. Görz u. a. (2003) geht tiefer auf die vorher genannten Algorithmen ein.

2.4.2 Fazit

Constraints bietet einen flexiblen Mechanismus um ein komplexes Konfigurationsproblem wie das vorliegende zu repräsentieren und zu lösen. Es existieren verschiedene Arten von Algorithmen, welche einen Kompromiss aus Effizienz und Qualität auch für schwer lösbare Probleme bereitstellen.

2.5 CSCW - Computer Supported Collaborative Work

Der Begriff des Computer Supported Collaborative Work wurde 1984 von Greif und Cashman geschaffen. CSCW umfasst Methoden und Technologien um Gruppenarbeit anhand rechnerseitigen Unterstützung zu realisieren. Der Bereich des Computer Supported Collaborative Work ist ein interdisziplinärer Forschungsbereich zwischen Informatik, Psychologie, Soziologie und weiteren. Der Begriff Groupware wurde ebenso in den frühen 80er Jahren von Johnson-Lenz geprägt. Groupware umfasst die Systeme, die Gruppenarbeit unterstützten. Da der eben genannte Groupware-Begriff sich von den mittlerweile bekannten Groupware-Lösungen (wie z.B. Lotus-Notes) unterscheidet, sollen Anwendungen die in den Bereich der Groupware fallen CSCW-Systeme genannt werden. CSCW-Systeme unterscheiden sich dabei von den vorher genannten Groupware-Lösungen in dem Sinne, dass CSCW-Systeme eine wirklich gemeinsame Umgebung bereitstellen. Sogenannte Groupwarelösungen stellen oft Funktionalitäten für Kontakt-, Kalender- und Projektmanagement zur Verfügung und sind ein Teilbereich der Groupware (Vgl. [Borghoff U. \(1998\)](#) S. 92-93).

Weiterhin existiert bei CSCW-Systemen eine Forderung nach einem Gruppenbewußtsein. Dies bedeutet, dass den einzelnen Gruppenteilnehmer bewußt ist, dass sie in einer Gruppe arbeiten und dies aktiv vom System gefördert wird. Z.B. könnte in einem Mehrautorensystem ein Benutzer sehen, welcher seiner Mitbenutzer gerade ebenso an einem Dokument arbeitet und automatisch über Änderungen informiert werden. Die Forderung nach einem Gruppenbewußtsein wäre hingegen nicht erfüllt, wenn die Benutzer nacheinander das Dokument bearbeiten und ihre Aktionen voneinander vollständig isoliert ablaufen (Vgl. [Borghoff U. \(1998\)](#) S. 96).

2.5.1 Klassifizierung von CSCW-Systemen

CSCW-Systeme können entweder nach Raum und Zeit oder nach dem sogenannten 3K-Modell klassifiziert werden. Nach [O'Hara-Devereaux M. \(1994\)](#) können CSCW-Systeme in vier Kategorien innerhalb der Raum-Zeit-Matrix eingeordnet werden. Diese Kategorien bestimmen sich aus Ort (gleich oder verschieden) und Zeit (gleich oder verschieden). ([Grudin \(1994\)](#), S.19 - 26) erweitert die Matrix um den Begriff der Vorhersehbarkeit. Abbildung 2.2 verdeutlicht die Einordnung und enthält jeweils ein Beispiel für die betreffende Kategorie.

2.5.2 3K-Modell

[Teufel S. \(1995\)](#) klassifiziert ein CSCW-System anhand dessen Anwendungsschwerpunkte hinsichtlich Kommunikation, Koordination und Kooperation innerhalb des 3K-Modells. Die Schwerpunkte der drei Bereiche lauten wie folgt:

Raum/Zeit	gleiche Zeit (synchron)	verschiedene Zeit (asynchron)	
		vorhersehbar	nicht vorhersehbar
gleicher Ort	<i>face-to-face</i> - Sitzung	Schichtarbeit	„schwarzes Brett“
verschiedener Ort vorhersehbar	Video- konferenz	E-mail	kollaboratives Verfassen von Dokumenten
verschiedener Ort nicht vorhersehbar	Mobilfunk- konferenz	Nicht-Realzeit- rechnerkonferenz	Vorgangs- bearbeitung

Abbildung 2.2: Klassifizierung nach Raum/Zeit (Quelle: [Borghoff U. \(1998\)](#) S. 121)

Kommunikation Verständigung zwischen Personen.

Koordination Effiziente Organisation von Ressourcen und Tätigkeiten.

Kooperation Unterstützung der Benutzer bei der Verfolgung eines gemeinsamen Zieles.

Diese „K-Orientierung“ lassen sich innerhalb eines Dreiecks wie in [Abbildung 2.3](#) veranschaulicht positionieren.

2.5.3 Fazit

Die Anwendung in dem vorgestellten Szenario aus [2.1](#) kann nach der Raum-Zeit-Klassifikation in „Raum verschieden vorhersehbar“ und „Zeit gleich,“ eingeordnet werden. Hinsichtlich des 3K-Modells erfolgt die Positionierung zwischen den beiden Ecken „Kooperation“ und „Kommunikation“, da das gemeinsame Arbeiten an einem Formular zusätzlich durch eine Webkonferenz unterstützt werden soll.

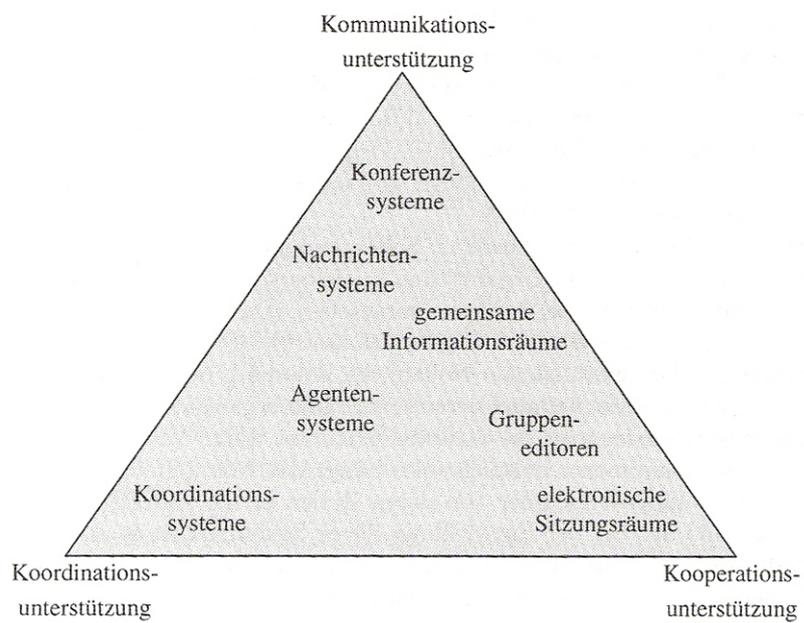


Abbildung 2.3: 3K-Modell (Quelle: [Borghoff U. \(1998\)](#)) S.128

3 Aktuelle Produkte und Arbeiten

Nach dem Wissen des Autors existiert kein Produkt, welches ein constraint-basiertes, kollaboratives Konfigurationssystem wie in dem vorhergehendem Kapitel beschrieben vereint. Daher sollen nun Systeme aus dem Bereich der Kollaboration und der Constraints auf eine mögliche Verwendung untersucht werden.

3.1 Kollaborations-Systeme

3.1.1 CommSy

Nach ([Jackewitz \(2005\)](#) S.32) ist CommSy ein webbasiertes Kooperationssystem und zielt auf die Unterstützung des Studiums an deutschen Hochschulen hin. Dabei hat es sich von einem projektorientierten Unterstützungssystem zu einer fach- und studienübergreifenden Plattform der universitären Lehre weiterentwickelt.

Aufbau

Auf der CommSy-Projektseite ¹ findet sich eine Beschreibung der Features von Commsy. Dabei existieren verschiedene virtuelle Räume innerhalb von CommSy:

Projektraum Ein Projektraum ist nur den Projektteilnehmern zugänglich und dient als Unterstützung zur Erarbeitung eines gemeinsamen Projektes. Dort können unter anderem relevante Termine verwaltet, Arbeitsmaterialien hinterlegt und erarbeitet, asynchrone Diskussionen geführt, Themen und Aufgaben verwaltet und gechattet werden.

Gemeinschaftsraum Der Gemeinschaftsraum ist einer größeren Teilnehmerzahl zugänglich als der Projektraum. Er dient zur Bündelung und Strukturierung von mehreren Projekten.

¹<http://www.commsy.net/Software/Features>

persönlicher Raum Jeder Teilnehmer besitzt einen persönlichen Raum, der nur ihm selbst zugänglich ist. Dort finden sich Informationen über die Tätigkeiten des Teilnehmers und die damit verknüpften Informationen wie z.B. Materialien, Gruppen oder Projekte. Zusätzlich kann der Teilnehmer hier Termine pflegen.

Eine weitere interessante Funktion ist die Umsetzung von CommSy als ein semantisches Web. Innerhalb eines semantischen Webs sind die verschiedenen Elemente, die miteinander in Beziehung stehen, verknüpft. Z.B. ist es nachvollziehbar, welche Materialien in welchem Projektraum in welcher Diskussionsrunde genutzt wurden. Ebenso kann geprüft werden wer an diesen Diskussionen teilgenommen hat und an welchen anderen Projekten und Diskussionsrunden sich die entsprechenden Teilnehmer beteiligen. Es entsteht also ein Netz aus Informationen, ein semantisches Web.

Die Dissertation von [Jackewitz \(2005\)](#) beschäftigt sich eingehender mit dem Thema CommSy.

Fazit

CommSy soll als Grundlage für das zu entwickelnde constraint-basierte kollaborative Konfigurationssystem dienen. Es bietet bereits Möglichkeiten Diskussionen asynchron zu führen, Informationen, Kontakte oder Termine zu verwalten, ein Rechtssystem und einiges mehr. Eine Komponente, die eine synchrone Zusammenarbeit an einem Dokument oder einem Formular ermöglicht existiert allerdings noch nicht und müsste als eine Erweiterung für CommSy implementiert werden. Auf die Art und Weise einer Implementation soll in der Ausarbeitung im Rahmen des AW2-Seminars eingegangen werden.

3.2 Constraint-Systeme

Anschließend sollen drei Constraint Systeme, die Java Constraint Library, die Java Expert System Shell und der ILog JSolver kurz vorgestellt werden.

3.2.1 JCL

Nach (vgl. [Torrens u. a. \(1997\)](#) [zitiert nach [Runte \(2006\)](#) S. 70]) ist die Java Constraint Library (JCL)² im Rahmen einer Diplomarbeit von Marc Torrens entstanden. Die Bibliothek ist OpenSource und steht unter LGPL und kann auch kommerziell eingesetzt werden. JCL

²<http://liawww.epfl.ch/JCL/>

unterstützt verschiedene Suchverfahren welche teilweise mit Verfahren zur Herstellung lokaler Konsistenzen kombiniert werden können. Ein Constraint-Erfüllungsproblem muss in einer JCL-spezifischen Sprachen, der CSP Description Language (CSPDL), definiert werden. Zusätzlich existiert eine grafische Oberfläche, welche das Editieren und Lösen von Constraint-Erfüllungsproblemen erlaubt. Laut (Runte (2006) S. 71) unterstützt JCL nur Constraints mit endlichen Wertebereichen mit maximal zwei Variablen, Constraints mit mehr Elementen müssen selbst reduziert werden. Zusätzlich bietet JCL keine Möglichkeit zur inkrementellen Constraint-Verarbeitung. Torrens u. a. (1997) beschreibt in seinem Paper die Nutzung von JCL an einem Beispiel.

3.2.2 ILog JSolver

Die Firma ILOG ist nach (vgl. Freuder und Wallace (2000) S. 20 - 23 [zitiert nach Runte (2006) S. 69]) erfolgreich im Bereich des kommerziellen Marktes von Constraint-Werkzeugen. Der ILog JSolver³ stellt die Java Variante für den ursprünglich für C++ ausgelegten ILog Solver. Beide sind Komponenten der ILOG OPTIMIZATION SUITE. Zum einen lassen sich mit dem ILog JSolver Constraints mit endlichen Wertebereichen nutzen und zum anderen wird der Aufbau von inkrementellen Constraint-Netzen unterstützt. Da es sich bei dem ILOG JSolver um ein kommerzielles Produkt handelt, fallen Nutzungskosten im fünfstelligen Bereich an.

3.2.3 JESS

Die Java Expert System Shell⁴ wurde von Ernest Friedman-Hill an den Sandia National Laboratories entwickelt. Es handelt sich dabei um eine deklarative Sprache, ähnlich wie LISP und kann für akademische Zwecke frei genutzt werden. Aus JESS können Java Objekte aufgerufen und über Java JESS-Code erweitert werden. Bei JESS handelt es sich nicht direkt um ein Constraint-System, sondern um ein Regelsystem, welches die Constraints erzeugt. Durch diesen Mechanismus unterstützt JESS indirekt inkrementelle Constraint-Netze. Eine ausführliche Behandlung von JESS wird in Friedman-Hill (2003) vorgenommen.

3.2.4 Fazit

Da JCL weder inkrementelle Constraint-Netze und Constraints mit endlichen Wertebereichen ordentlich unterstützt scheidet es bereits im Vorfeld aus. JESS erfordert unter anderem eine

³<http://www.ilog.com.sg/products/jsolver/>

⁴<http://www.jessrules.com/>

Integration des Regelsystems nach Java und unterliegt dadurch gewissen Einschränkungen. Da nach Wissen des Autors die HAW eine entsprechende Lizenz für den ILog JSolver besitzt, kann der ILog JSolver im Rahmen einer Ausarbeitung eingesetzt werden. Eine tiefer gehende Prüfung soll im Rahmen des AW2-Seminars erfolgen.

3.3 verwandte Arbeiten

Innerhalb der Arbeiten welche innerhalb der HAW erstellt wurden sind bezüglich der Problemstellung zwei Exemplare interessant:

Berechtigungsstrukturen in kollaborativen Umgebungen von Horst Mund ([H. \(2006\)](#))

Ein System für das kollaborative Bearbeiten von Dokumenten in mobilen Umgebungen
von Alexandra Revout ([Revout \(2008\)](#))

Horst Mund beschäftigt sich mit Rechtestrukturen innerhalb einer kollaborativen Umgebung und Alexandra Revout beschreibt ein verteiltes asynchrones Dokumentensystem.

4 Ausblick und Risiken

In diesem Kapitel soll anschließend ein Ausblick auf die folgende Vorgehensweise gegeben und mögliche Risiken aufgezeigt werden.

4.1 Ausblick

Innerhalb des AW2-Seminars sollen auf Basis dieser Ausarbeitung folgende Schritte durchgeführt werden:

Szenario festlegen :

- Erarbeitung des Expertenwissens
- Definition der nötigen Konfigurationsobjekte und deren Lösungsräume
- Auswahl eines Constraint Systems

Technologieprüfung :

- Schnittstelle zu CommSy (Plugin)
- Constraint System auf Tauglichkeit prüfen
- Webtechnologie für Umsetzung finden

4.2 Risiken

Während der Bearbeitung im Rahmen des AW2-Seminars können folgende Risiken auftreten:

- Expertenwissen nicht zugänglich
- Erweiterung von CommSy nicht möglich
- Constraint System unzureichend
- Webtechnologie erfüllt nicht alle Bedingungen

Bei der Erarbeitung des Szenarios ist es möglich, dass das entsprechende Expertenwissen nicht wie notwendig zugänglich ist. In dem hier vorgestellten Szenario aus dem Bereich des Steuerrechts scheint dies sehr wahrscheinlich. Zusätzlich ist möglicherweise eine Erweiterung von CommSy nicht in dem vorgestellten Rahmen möglich. Weiterhin muss das Constraint-System auf seine Tauglichkeit geprüft werden. Unabhängig von den einzelnen Produkten muss die gewählte Webtechnologie die mit CommSy und dem Constraint-System zusammenarbeiten und eine Webkonferenz ermöglichen.

Literaturverzeichnis

- [Brauer:1987 1987] : *Wissensbasierte Systeme : 2. Internationaler GI-Kongreß ;München, Oktober 1987 ; Proceedings*. Berlin [u.a.] : Springer, 1987. – ISBN 3-540-18494-5
- [Borghoff U. 1998] BORGHOFF U., Schlichter J.: *Rechnergestützte Gruppenarbeit*. Springer-Verlag, 1998
- [Cunis und Günter 1991] CUNIS, Roman ; GÜNTER, Andreas: *PLAKON - Übersicht über das System*. 1991
- [Freuder und Wallace 2000] FREUDER, Eugene C. ; WALLACE, Mark: *Constraint Technology and the Commercial World (Interview)*. In: *IEEE Intelligent Systems 15 Januar/Februar*. 2000
- [Friedman-Hill 2003] FRIEDMAN-HILL, Ernest: *Jess in Action - Java Rule-based Systems*. Manning Publications Co., 2003
- [Garey und Johnson 1979] GAREY, Michael R. ; JOHNSON, David S.: *Computers and Intractability - A Guide to the Theory of NP-Completeness*. (1979)
- [Günter 1991] GÜNTER, Andreas: *Expertensysteme für Konstruktionsaufgaben*. In: ?? (1991)
- [Günther und Kühn 1999] GÜNTHER, Andreas ; KÜHN, Christian: *Knowledge Based Configuration - Survey and Future Directions*. In: *Knowledge-Based Systems - Survey and Future Directions, Proceedings of the 5th Biannual German Conference on Knowledge-Based Systems (XPS 99)*, 1999
- [Grudin 1994] GRUDIN: *CSCW: History and Focus*. *IEEE Computer*. 1994. – 19–26 S
- [Görz u. a. 2003] GÖRZ, Günther ; ROLLINGER, Claus-Reiner ; SCHNEEBERGER, Josef: *Handbuch der künstlichen Intelligenz*. Oldenbourg Verlag München Wien, 2003
- [H. 2006] H., Mund: *Berechtigungsstrukturen in kollaborativen Umgebungen*, HAW-Hamburg, Diplomarbeit, 2006
- [Hentenryck und Saraswat 1996] HENTENRYCK, Pascal V. ; SARASWAT, Vijay: *Strategic directions in constraint programming*. 1996

- [Jackewitz 2005] JACKEWITZ, Iver: *Evolutionary Application Service Providing (eASP) Ein Ansatz der Softwarebereitstellung*, Universität Hamburg, Dissertation, 2005
- [O'Hara-Devereaux M. 1994] O'HARA-DEVEREAUX M., Johansen R.: *GlobalWork: Bridging Distance, Culture and Time*. Jossey-Bass Inc. San Francisco, 1994
- [Revout 2008] REVOUT, Alexandra: *Ein System für das kollaborative Bearbeiten von Dokumenten in mobilen Umgebungen*, HAW-Hamburg, Diplomarbeit, 2008
- [Runte 2006] RUNTE, Wolfgang: *YACS: Ein hybrides Framework für Constraint-Solver zur Unterstützung wissensbasierter Konfigurierung*, Universität Bremen, Diplomarbeit, 2006
- [Teufel S. 1995] TEUFEL S., Sauter C.: *Computerunterstützte Gruppenarbeit*. Addison-Weasly, 1995
- [Torrens u. a. 1997] TORRENS, Marc ; WEIGEL, Rainer ; FALTINGS, Boi: Java Constraint Library: bringing constraints technology on the Internet using the Java language. In: *Workshop on Constraints and Agents (AAAI97)* Bd. Technical Report WS-97-05, AAAI Press, 1997