

Universität Bamberg
Angewandte Informatik

Seminar KI: gestern, heute, morgen

Worauf muss bei der Entwicklung eines CBR-Systemes
geachtet werden?

von

Katrin Gisela Maria Grabe

25.02.2016

betreut von
Ute Schmid, Dietrich Wolters

Abstract

Fallbasiertes Schließen (Case Based Reasoning CBR) ist eine Form des Problemlösens, bei dem die Lösung eines spezifischen vorher bereits gelösten Falls auf das neue Problem angewandt wird. In dieser Arbeit wird der Frage nachgegangen, welche Schritte eines CBR-Systems die wichtigsten sind, sodass auf sie bei der Entwicklung besonders geachtet werden muss. Erklärt wird zunächst allgemein der CBR-Zyklus, anschließend wird auf das erste CBR-System, den Mediator von Kolodner und Simpson eingegangen. Dazu werden vorher die Methoden des Analogieschließens und des vernunftbasierten Schließens (Commonsense Reasoning) vorgestellt, um dann das Vorgehen des Mediators an einem Beispielfall erklären zu können. Anhand des allgemeinen CBR-Zyklus und des Mediators wird abschließend die Forschungsfrage nach den wichtigsten Bestandteilen eines CBR-Systems mit der Fallbasis und dem Indexvokabular und Lösungen für die Recieve-, Reuse-, Revise- und Retain-Phasen beantwortet.

Inhaltsverzeichnis

Abstract	I
1 Einleitung	2
2 Grundkonzept des fallbasierten Schließens	2
2.1 Definition	2
2.2 CBR-Zyklus	3
2.3 Fallbasis	4
2.4 Retrieve	4
2.5 Reuse	5
2.6 Revise	5
2.7 Retain	6
3 Beispiel Mediator	6
3.1 Analogieschließen	7
3.2 Vernunftbasiertes Schließen	7
3.3 Vorgehensweise des Mediators	7
3.4 Mediator: Fallrepräsentation	8
3.5 Mediator: Retrieve	10
3.6 Mediator: Reuse	11
3.7 Mediator: Retain	11
4 Zusammenfassung und Fazit	12
Literatur	13

1 Einleitung

Ein Patient kommt zum Arzt. Kein Anfang eines schlechten Witzes, sondern der Anfang eines alltäglichen Prozesses. Nach mehr oder weniger gründlicher Untersuchung schlägt der Arzt im Normalfall nicht sein Lehrbuch auf, sondern ruft im Geiste Patienten auf, die mit ähnlichen Symptomen vor ihm gesessen haben. Ortswechsel in das Hinterzimmer eines Gerichtssaales. Ein Richter grübelt nach der Anhörung aller Beteiligten über ein angemessenes Urteil für den Angeklagten. Auch er erinnert sich an frühere Fälle und ihre Urteile. Zoom-in zum Laptop des sich in der Pause befindenden Prozessprotokollanten. Dieser surft auf Verkaufsportalen, immer verfolgt von dem Kästchen "Kunden, die dieses Produkt kauften, interessierten sich auch für...". Was ihm nicht bewusst ist, diesen Werbeblock erstellte der Online-Shop-Betreiber durch einen Abgleich dieses Kunden mit der Käuferdatenbank. In Medizin und Rechtsprechung wird es bereits länger angewandt, Online-Versandhändler nutzen es seit einigen Jahren, das fallbasierte Schließen. Ein aktuelles spezifisches Problem liegt vor, die Aufgabe ist es, eine konkrete Lösung zu finden. Nach der Problem-Untersuchung bedient sich das fallbasierte Schließen an einer Datenbank früherer Probleme und sucht den ähnlichsten Einzelfall heraus. Anhand seiner Lösung wird eine möglichst zufriedenstellende Lösung für den aktuellen Fall entwickelt und evaluiert. Anschließend wird der aktuelle Fall in die Datenbank aufgenommen. Im Groben fasst dies den Vorgang des fallbasierten Schließens (kurz CBR aus dem Englischen Case-Based-Reasoning) zusammen. Beim genaueren Untersuchen eines CBR-Systems werden seine Vor- und Nachteile deutlich, sodass sich die Frage stellt: Worauf muss bei der Entwicklung eines CBR-Systems geachtet werden? Dies ist die Forschungsfrage dieser Arbeit. Dafür wird, nach der kurzen Einführung in diesem ersten Kapitel, im zweiten Kapitel erläutert, wie CBR-Systeme arbeiten. Das dritte Kapitel befasst sich mit einem konkreten Beispiel aus der Praxis und zwar mit dem prominentesten Fall dieses Bereichs, der Lösung des Sinai-Konfliktes durch J. Kolodners und R. Simpsons Mediator (Kolodner and Simpson (1989)). Insbesondere wird auf Stärken und Schwächen dieses Systems geachtet, sodass im Fazit, im vierten Kapitel, die Forschungsfrage beantwortet werden kann und Teile des Systems benannt werden, auf die bei der Entwicklung besonders geachtet werden muss.

2 Grundkonzept des fallbasierten Schließens

2.1 Definition

Das Fallbasierte Schließen basiert auf zwei Annahmen:

1. "Ähnliche Probleme haben ähnliche Lösungen." (Beierle and Kern-Isberner, 2014, S. 160)
2. "Jedes Problem ist anders, aber der Typ der Aufgabenstellung wiederholt sich." (Beierle and Kern-Isberner, 2014, S. 160)

Mit dieser Grundlage kann das fallbasierte Schließen präzise definiert werden: Fallbasiertes Schließen ist das Entwickeln angepasster Lösungen für neue Probleme aus einer

Datenbank bereits gespeicherter Fälle. Es wird aus einem spezifischen Fall eine neue spezifische Lösung für einen neuen spezifischen Fall entwickelt. (Beierle and Kern-Isberner, 2014, S. 158-159)

2.2 CBR-Zyklus

In der künstlichen Intelligenz wird das fallbasierte Schließen mit Hilfe eines Computersystems ausgeführt. Diese Systeme sind individuell entwickelbar und unterscheiden sich inhaltlich. Der grobe Ablauf des fallbasierten Schließens kann aber im Case-Based-Reasoning-Zyklus (kurz CBR-Zyklus) zusammengefasst werden (siehe Abbildung 1) und wird im Folgenden erklärt.

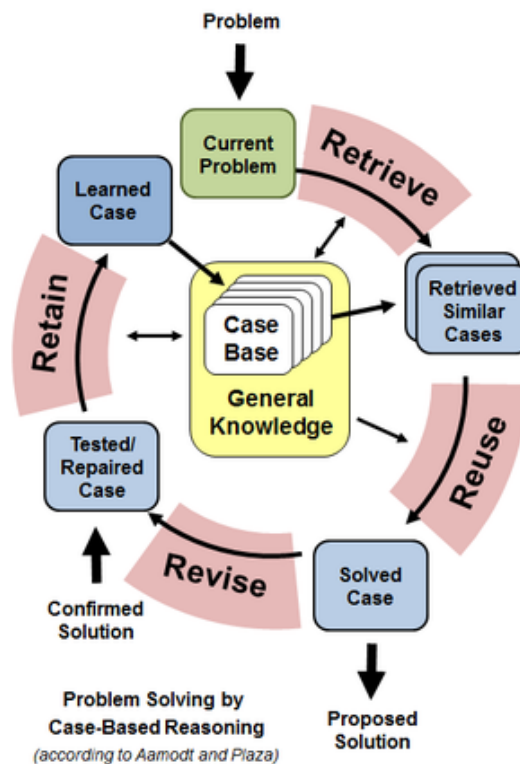


Abbildung 1: CBR-Zyklus

Zu Rate gezogen wird ein CBR-System, wenn ein konkretes Problem besteht, das anhand der Anpassung der Lösung eines früheren Falls gelöst werden soll. Der CBR-Vorgang startet mit dem Aufnehmen des aktuellen Problems in das System (in Abbildung 1 grünes Feld). Auf unterschiedliche Weise je nach System werden die Indexwerte des aktuellen Falls z.B. durch eine Nutzereingabe an das System übergeben, damit beginnt die Retrieve-Phase. In dieser ersten Phase vergleicht das CBR-System anhand der Werte des Indexvokabulars den aktuellen Fall mit seiner Fallbasis und erkennt anhand der Grobsuche alle passenden Fälle. In der anschließenden Feinsuche ermittelt das System

intern aus den Ergebnissen der ersten Suche den am besten passenden Fall. Mit diesem Fall wird weitergearbeitet. Es werden die Unterschiede zwischen dem Ergebnis und dem aktuellen Problem ermittelt und unterschiedliche Parameter angepasst, die Reuse-Phase. Anschließend wird die Lösung des alten Falls abgeändert und als neue Lösung des aktuellen Problems vorgeschlagen. Anhand einer Prüfung durch das System selbst oder durch den anfragenden Nutzer wird die Lösung angenommen bzw. verworfen und die Qualität der Lösung in der Revise-Phase getestet. Eventuell müssen Anpassungen vorgenommen werden. Wichtig ist, dass ein CBR-System ein lernendes System ist und durch die vierte Phase, die Retain-Phase, das aktuelle Problem mit seiner angepassten Lösung in seine Fallbasis aufnimmt, sodass es bei weiteren Problemen als Lösungsvorschlag dienen kann. (Beierle and Kern-Isberner, 2014, S. 165-167)

2.3 Fallbasis

Das Herzstück jedes CBR-Systems ist die Fallbasis, eine Datenbank, die alle Einzelfälle des Systems beinhaltet. Gespeichert werden sie in der Form, die das Indexvokabular vorgibt, jeder Fall ist also mit einer eindeutigen Kombination an Idizien identifizierbar. Ergänzt wird die Fallbasis um allgemeines meist bereichsabhängiges Wissen wie z.B. die Standard-Körperwerte PAT bei einem medizinischen System. Eine optimale Fallbasis ist möglichst groß mit prägnanten Fällen, die durch präzises Indexvokabular gut voneinander abgrenzbar sind. Wenn es viele ähnliche oder sich wiederholende Fälle gibt, wirkt sich das nicht gut auf die Leistung eines Systems aus, wohingegen nur sehr wenige Fälle keine gute Ausgangslage für verschiedene anfragende Probleme sind. Größe der Fallbasis und Präzision des Indexvokabular sind Stellschrauben für die Qualität eines CBR-Systems und somit Teil der Antwort auf die Forschungsfrage dieses Textes. Das Indexvokabular setzt sich aus einer Menge von Attributen und der jeweils dazugehörigen Wertemenge zusammen. Gute Deskriptoren folgen der Terminologie des thematischen Gebiets und benutzen Begriffe, die bei der Fallselektion benutzt werden. Außerdem müssen sie die Balance zwischen hinreichend abstrakt und hinreichend konkret finden. Inhaltlich decken Sie den Kontext, das Problem, die Lösung und Resultate der Lösungsdurchführung ab. (Beierle and Kern-Isberner, 2014, S. 176-177)

2.4 Retrieve

Eine treffende Konstruktion des Indexvokabulars ist die Voraussetzung für das Finden des am besten zum aktuellen Fall passenden Falls der Fallbasis. In der Retrieve-Phase steht das System vor einem Suchproblem und einem Ähnlichkeitsproblem. Auch wenn das Indexvokabular die entscheidenden Parameter der Einzelfälle beinhaltet, wird jetzt ein Algorithmus benötigt, der während der Retrieve-Phase die zueinander passenden Fälle findet. Die Mehrzahl der CBR-Systeme errechnet den besten Fall der Fallbasis mit Ähnlichkeitsberechnungen durch Techniken wie z.B. Entscheidungsbäumen (Baydin et al., 2011, S. 386). Dazu wird die Ähnlichkeit zwischen dem aktuellen Fall und den möglicherweise passenden Fällen der Fallbasis, die durch die Grobsuche ermittelt wurden, berechnet. Für die Ähnlichkeitsberechnung gibt es verschiedene Ähnlichkeitsmaße

wie die (gewichtete) Hamming-Ähnlichkeit, die verallgemeinerte Ähnlichkeit und die PATDEX/2-Ähnlichkeit (genauer nachzulesen bei Beierle und Kern-Isberner (Beierle and Kern-Isberner, 2014, S. 186-197)).

2.5 Reuse

Nach der Auswahl eines Falls als den besten für das aktuelle Problem ist es an dem System, die alte Lösung an die neuen Gegebenheiten anzupassen. Dazu gibt es verschiedene Vorgehensweisen siehe Abbildung 2

- Substitutionsmethoden
 - Reinstantiierung
 - Lokale Suche
 - Parameteranpassung
 - Fallbasierte Substitution
- Transformationsmethoden
 - Common-sense Transformation
 - Modellbasierte Transformation
- Spezielle Adaption und Verbesserung
- Derivationswiederholung

Abbildung 2: Adaptationsmethoden

Bei der Substitution wird die Lösung des Referenzfalls genommen und ihre Bestandteile durch Teile des neuen Falls ersetzt. Die Transformation wandelt die Lösung als Ganzes um, sodass die Struktur der Lösung verändert wird, z .B. wenn ein Bestandteil beim aktuellen Fall weggelassen wird. Spezielle Adaption und Verbesserung steht für Heuristiken, die keiner anderen Kategorie zugeordnet werden können. Insbesondere fallen Methoden, die beim Fehlschlagen einer Lösung zur Korrektur angewendet werden, in diesen Bereich. Die letzte Kategorie der Derivationswiederholung ist eine Ableitung von neuen Werten aus der alten Methode. Ein Beispiel ist das Lösen einer Rechenaufgabe. Eine neue Aufgabenstellung wird gelöst, in dem der Lösungsweg einer alten Aufgabe nachvollzogen und angepasst wird. Hier wurde nur ein kurzer Überblick über verschiedene Adaptationsmethoden gegeben, genauere Einblicke in Beierle und Kern-Isberner (Beierle and Kern-Isberner, 2014, S. 197-201).

2.6 Revise

Der gelöste Fall wird dem Nutzer, der auch aus mehreren Personen oder den Konfliktparteien bestehen kann, vorgeschlagen. Akzeptiert dieser die Lösung nicht, wird im Kreislauf

zurückgegangen und Änderungen in Adaptation, Fallauswahl oder sogar in der Falleingabe vorgenommen. Dieses Zurückgehen und Anpassen kann theoretisch unbegrenzt oft passieren. Wurde die Lösung des Falls durch den Nutzer angenommen, wurde bereits der erste Test der Retrieve-Phase bestanden. Das Testen der Lösung ist von CBR-System zu CBR-System unterschiedlich und somit wieder ein Qualitätsfaktor zur Beantwortung der Forschungsfrage. Optimal ist es, wenn die Lösung erst in der Praxis ausprobiert wird, bevor der Fall in die Fallbasis aufgenommen wird. So kann auch das Gelingen oder Fehlschlagen der Lösung dokumentiert werden. Ansonsten kann das Testen sowohl durch den Nutzer (z.B. durch Einschätzung) als auch durch das System (z.B. durch Simulation) stattfinden (Beierle and Kern-Isberner, 2014, S. 201-202). Der Spielraum für den Systementwickler liegt hier zwischen einem vollautomatischen System, das durch Interaktionen lediglich Feedback vom Nutzer für die durch das System getroffenen Entscheidungen bekommt. Der Gegenpol ist ein retrieval-only-System, das Fälle zwar bereitstellt, aber nur um das Gedächtnis der Nutzer zu ergänzen, Entscheidungen werden vom Nutzer getroffen. (Kolodner, 1992, S. 30-31)

2.7 Retain

Negative Ergebnisse beim Testen sorgen immer für eine Verbesserung des Systems z.B. wenn sich daraus ein Nachjustieren des Indexvokabulars ergibt, damit in Zukunft direkt der am besten passende Fall gefunden wird. Außerdem werden aufgetretene Probleme und Nachjustierungen direkt in die Beschreibung des aktuellen Falls mit aufgenommen, sodass beim erneuten Verwenden dieses Falls explizit auf das Vermeiden dieser geachtet werden kann. Eine zweite Art der Indexanpassung, die von einigen CBR-Systemen genutzt wird, ist die Generalisation. Wenn mehrere Fälle mit den gleichen Parametern auf den Indizes die gleiche Lösung vorschlagen, können sie zu einem Fall zusammengefasst werden. Dem Indexvokabular wird dann je nach System eine Klasse oder eine Auflistung an Beispielen hinzugefügt, um die einzelnen Fälle nicht zu verlieren (Kolodner, 1992, S. 19-20). Unabhängig vom Testergebnis ist der letzte Schritt des CBR-Zyklus die Aufnahme des aktuellen Falls in die Fallbasis. Dazu wird zunächst überprüft, ob der Unterschied zwischen dem aktuellen Fall und dem Referenzfall groß genug ist, sodass das System qualitativ ergänzt wird. Fällt diese Prüfung positiv aus, werden vorher fehlende Indizes ergänzt und der Fall in der Fallbasis abgespeichert. (Beierle and Kern-Isberner, 2014, S.201-202)

3 Beispiel Mediator

Der Mediator ist ein fallbasiertes System zum Problemlösen, das eingesetzt wird um Ressourcenkonflikte zu lösen. 1985 wurde er von J. Kolodner und R. Simpson als eines der ersten CBR-Systeme entwickelt (Weiterentwicklungen von anderen KI'lern sind CHEF, CASEY, JULIA, weitere Informationen unter Kolodner and Simpson (1989).) Zur Vorbereitung auf den Mediator müssen zwei Themenbereiche der künstlichen Intelligenz kurz dargestellt werden, die in diesem System Anwendung finden, das Analogieschließen und das vernunftbasierte Schließen (commonsense reasoning).

3.1 Analogieschließen

Das Analogieschließen hat als Ausgangslage eine Zieldomäne, für die es ein Problem zu lösen gilt. Außerdem ist eine Basisdomäne gegeben, die ein ähnliches Problem bereits gelöst hat. Um das Problem der Zieldomäne zu lösen, muss der Lösungsvorgang der Basisdomäne abstrahiert werden. Dann wird die Zieldomäne auseinandergenommen, sodass ihre Elemente in den abstrahierten Fall der Basisdomäne eingesetzt werden können. Ein berühmtes Beispiel dafür ist die Übertragung der Elemente des heliozentrischen Modells auf das Atommodell durch Rutherford. Die Gleichsetzung findet statt zwischen Sonne und Atomkern, Planeten und Elektronen und Planetenbahnen und Atomschalen. Dörner (1980) Das Analogieschließen steht in enger Verwandtschaft zum fallbasierten Schließen. Beide Methoden lösen neue Probleme, indem sie sich auf bekannte Fälle berufen. Der Unterschied besteht in der Umgangsweise mit dem Referenzfall. Beim Analogieschließen wird abstrahiert oder nach einer strukturellen Ähnlichkeit gesucht. Beim fallbasierten Schließen wird durch die Indizes auf Beziehungen zwischen den beiden speziellen Fällen eingegangen.

3.2 Vernunftbasiertes Schließen

Das vernunftbasierte Schließen war in der künstlichen Intelligenz lange Zeit die Fähigkeit, die ein System beherrschen musste, um tatsächlich intelligent zu sein. Es ist der Vorgang Informationen aus dem täglichen Leben zu verstehen und zu manipulieren. Ein Beispiel ist die Erkennung von Objekten, die für sich gesehen schwierig zu erkennen wären, im Kontext aber durch Farbe, Größe oder Platzierung für einen Akteur mit menschlichem Hintergrundwissen leicht identifizierbar sind. Die Gemeinsamkeit des fallbasierten Schließens und des vernunftbasierten Schließens ist, dass es bei beiden um eine konkrete Situation geht. Beim fallbasierten Schließen wird aber im Unterschied zum vernunftbasierten Schließen ein aufwändiger Prozess (Suche des Referenzfalls, Anpassung, Feedback) durchlaufen. Das vernunftbasierte Schließen orientiert sich am menschlichen Denkvorgang, der schnell und intuitiv vorgeht (Davis and Marcus, 2015, S.92-105). Das AI-System ConceptNet des Massachusetts Institute of Technology (MIT) ist ein Open-Source-Projekt mit verschiedenen Komponenten. Für diese Arbeit ist vor allem die Funktion als Wissensbasis von Bedeutung, in der ConceptNet Informationen des alltäglichen Lebens sammelt, klassifiziert und in Form von Ontologin speichert. Ein anderes Beispiel ist WordNet, ein lexikalisch-semantisches Netz, das lexikalische und semantische Beziehungen zwischen Wörtern des menschlichen Alltags wie Synonyme, Hyper- und Hyponyme sammelt und speichert (Baydin et al., 2011, S. 380- 381).

3.3 Vorgehensweise des Mediators

Die Strategie des Mediators geht nach dem CBR-Kreislauf vor und nutzt dazu Analogieschließen und das vernunftbasierte Schließen. Das genaue Vorgehen ist in Abbildung 3 dargestellt.

Es kann grob in drei Schritte zusammengefasst werden. Zuerst wird eine Ontologie des aktuellen Problems definiert, in die die Sicht aller beteiligten Konfliktparteien mit

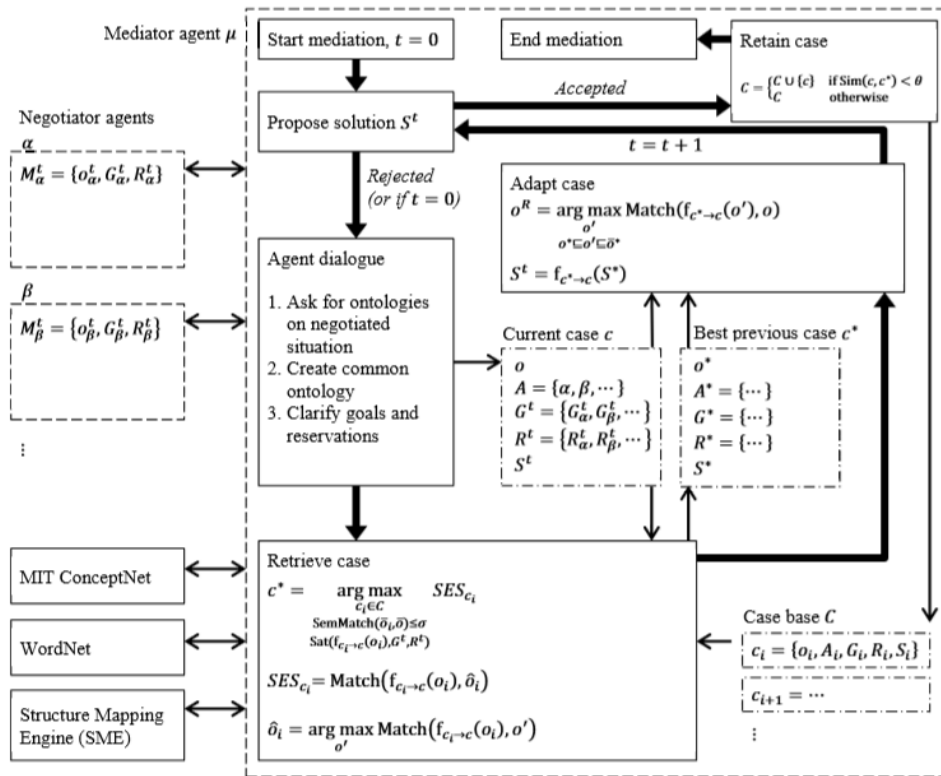


Abbildung 3: Mediator detailliert (Baydin et al., 2011, S. 383)

einfließt. Anschließend wird anhand dieses Falls durch Analogieschließen der ähnlichste Fall aus der Datenbank herausgesucht. Weil die Fälle oft zu unterschiedlichen Domänen gehören wird sich hier nicht des normalen Analogieschließens sondern des Structure Mapping Engines (SME) bedient, was Analogieschließen zwischen verschiedenen Domänen ist. Im dritten Schritt wird anhand der Ontologin für den aktuellen und den gefundenen Fall die Lösung des Alten an den Neuen angepasst, dazu wird das WordNet und das ConceptNet genutzt (Baydin et al., 2011, S. 382- 383). Beim näheren Erklären des Vorgehens des Mediators wird auf die kritischen Phasen eines CBR-Systems eingegangen und somit gleichzeitig auf die Forschungsfrage. Zur Veranschaulichung wird nach der Erklärung jedes Schritts das Vorgehen am Beispiel des Sinai-Konflikts erklärt, übersichtlich dargestellt ist dieser Mediationsprozess in Abbildung 4.

3.4 Mediator: Fallrepräsentation

Der Mediationsprozess wird gestartet, es ist der Zeitpunkt $t = 0$, in Abbildung 3 befinden wir uns oben links. Zunächst kann noch keine Lösung vorgeschlagen werden, weil der aktuelle Fall erst einmal aufgenommen werden muss. Dazu erstellt der Mediator im Dialog mit den Agenten (in der Abbildung α und β) eine Ontologie für den neuen Fall

1. CBR new case <i>New case c</i> <i>5 concepts, 6 relations</i> (Wants Egypt Sovereignty) (Wants Israel Security) (NeededFor Sinai Sovereignty) (NeededFor Sinai Security) (Gets* Egypt Sinai) (Gets* Israel Sinai)	1.b. Ontology expansion <i>Ontology expansion factor: 6.0</i> <i>30 concepts, 38 relations</i> (Wants Egypt Sovereignty) (Wants Israel Security) (NeededFor Sinai Sovereignty) (NeededFor Sinai Security) (Gets* Egypt Sinai) (Gets* Israel Sinai) (PartOf Civilian Sinai) (PartOf Military Sinai) ...
2. CBR retrieval <i>Best case c* from case base C (Total SME score: 67.28)</i> <i>26 possible analogies, average SME score: 2.58, maximum SME score: 3.90</i> (Desires Sister1 Cake) (Desires Sister2 Drink) (UsedFor Orange Cake) (UsedFor Orange Drink) (PartOf Peel Orange) (PartOf Pulp Orange) (UsedFor Peel Cake) (UsedFor Pulp Drink) (Gets Sister1 Peel) (Gets Sister2 Pulp) } <i>Solution S*</i>	<i>Best analogy for case c*</i> <i>Analogy 1 (SME score: 3.90)</i> Sister1 -> Egypt Sister2 -> Israel ... <i>25 other (rejected) analogies</i> <i>Analogy 2 (SME score: 2.70)</i> Pulp -> Israel Orange -> Middle East ... <i>Analogy 3 (SME score: 2.46)</i> Glass -> Sinai Drink -> Sovereignty
3. CBR adaptation <i>Best analogy f_{c*-c} (SME score: 3.90)</i> Sister1 -> Egypt Sister2 -> Israel Orange -> Sinai Cake -> Sovereignty Drink -> Security Peel -> Civilian Pulp -> Military ...	3.b. Inferences and solution (Gets Sister1 Peel) -> (Gets Egypt Civilian) (Gets Sister2 Pulp) -> (Gets Israel Military) ...
4. CBR retaining <i>New case c solved</i> (Wants Egypt Sovereignty) (Wants Israel Security) (NeededFor Sinai Sovereignty) (NeededFor Sinai Security) (PartOf Civilian Sinai) (PartOf Military Sinai)	(NeededFor Civilian Sovereignty) (NeededFor Military Security) (Gets Egypt Civilian) (Gets Israel Military) } <i>Solution S</i>

Abbildung 4: Mediator Sinai (Baydin et al., 2011, S. 384)

c_i in der Form

$$c_i = \{o_i, A_i, G_i, R_i, S_i\} \quad (1)$$

aus Ontologie(o_i), Agenten(A_i), Zielen(G_i), Einschränkung(R_i) und Lösungen (S_i) zusammengesetzt. Diese Kombination der Attribute ist spezifisch für den Mediator, beim Erstellen eines eigenen CBR-Systems kann sie frei gewählt werden.

Die Agenten übergeben die eigene Ontologie zum Fall, das Ziel und die Einschränkung in ähnlicher Form

$$M_\alpha^t = \{o_\alpha^t, G_\alpha^t, R_\alpha^t\} \quad (2)$$

(Baydin et al., 2011, S. 383- 385). In dem Sinai-Beispiel (Nr. 1) werden fünf Konzepte (Informationen), Ägypten, Israel, Souveränität, Sicherheit und Israel, in sechs Beziehungen gesetzt. Ägypten will Souveränität, Israel Sicherheit, dafür benötigen beide Israel. (Kolodner and Simpson, 1989, S. 511)

3.5 Mediator: Retrieve

In der Retrieve-Phase benutzt der Mediator die bereits erstellten Ontologien, das Analogieschließen über verschiedenen Domänen, SME, und vernunftbasiertes Schließen. Die Berechnung des am besten passenden Falls c^* wird im Algorithmus in Abbildung 5 dargestellt.

```

procedure RETRIEVE( $c, C$ )                                ▷ Current case  $c$ , case base  $C$ 
  for each case  $c_i$  in the case base  $C$  do
    Compute  $\hat{o}_i = \arg \max_{o' \sqsubseteq o} \text{Match}(f_{c_i \rightarrow c}(o_i), o')$   ▷ Expansions  $\hat{o}_i$  of ontology  $o$ 
    Compute  $SES_{c_i} = \text{Match}(f_{c_i \rightarrow c}(o_i), \hat{o}_i)$           ▷ Structural evaluation scores
  end for
  Select  $c^* = \arg \max_{\substack{c_i \in C \\ \text{SemMatch}(\hat{o}_i, \bar{o}) \leq \sigma \\ \text{Sat}(f_{c_i \rightarrow c}(o_i), G^t, R^t)}} SES_{c_i}$ 
  return  $c^*$                                              ▷ Case with best matching
end procedure

```

Abbildung 5: Retrieve-Phase (Baydin et al., 2011, S. 383)

Zwischen allen Fällen der Fallbasis C und dem aktuellen Fall c wird das Maximum des strukturellen Bewertungswerts (SES)

$$\text{Compute } o = \arg \max \text{Match}(f_{c_i \rightarrow c}(o_i), o_i) \quad (3)$$

und der SES selbst

$$\text{Compute } SES_{c_i} = \text{Match}(f_{c_i \rightarrow c}(o_i), o_i) \quad (4)$$

errechnet. Dabei ist f ein Analogiemapping von o_i von der Domäne c_i auf c und das zweite o_i wird durch die Formel (3) bestimmt, die sich des vernunftbasierten Schließens bedient. Dann wird der am besten passende Fall c^* aus diesen Fällen ausgewählt

$$\text{Select } c^* = \arg \max SES_{c_i} \quad (5)$$

Die Bewertung der Ähnlichkeit und somit die Auswahl des Referenzfalls ist eine weiteres Merkmal des Mediators. Alternativ kann ein anderes Ähnlichkeitsmaß gewählt oder das Nutzen der Ontologien durch das Nutzen von Baumstrukturen wie das ConceptNet ersetzt werden.

Für den Sinai-Konflikt (Nr. 2) ist der am besten passende Fall ein Streit zwischen zwei Schwestern um eine Orange. Die eine Schwester möchte einen Kuchen backen und benötigt dazu die Schale der Orange, die andere braucht das Fruchtfleisch der Frucht, um daraus ein Getränk herzustellen. Die Lösung ist ein ungleiches Teilen der Orange in Schale und Fruchtfleisch, sodass beide Schwestern den gewünschten Teil der Orange erhalten. Die Konzepte dieses Falls werden in Analogie gesetzt zu den Konzepten

des Sinai-Konflikts und es entstehen 26 verschiedene Kombinationsmöglichkeiten mit jeweils unterschiedlichen SME-Werten, also Werten der Übereinstimmung der verschiedenen kombinierten Konzepte.

3.6 Mediator: Reuse

Dieser Fall wird in der Reuse- oder Adaption-Phase durch substitutionelle Adaptation angepasst. Dies geschieht mit der Analogiemapping-Funktion f von c^* zu c

$$f_{c^* \rightarrow c}(S^*) \quad (6)$$

Dabei wird nach der Lösung (S) von c^* vorgegangen, die Parameter (G, A, R) von c^* durch die Parameter von c ersetzt. Adaptationsmethoden gibt es verschiedene, wie in Kapitel 2.5 deutlich wurde. Hier wird die Substitutionsmethode angewandt, diese bietet sich beim Analogieschließen an. An diesem Punkt wird der Zähler der Zeit t um einen Wert erhöht und die Lösung S^* den Agenten vorgeschlagen. Diese können die Lösung entweder verwerfen oder akzeptieren. Im Falle des Verwerfens wird ihr Ziel M_a^t angepasst indem die Ontologie (o), das Ziel (G) oder die Einschränkung (R) verändert werden.

Im Streit um die Insel Sinai (Nr.3) wird die Analogie mit dem höchsten SME-Wert ausgewählt. Die beiden Schwestern werden mit den Konfliktländern, die Orange mit der Insel, Kuchen und Getränk mit Souveränität und Sicherheit und die Schale und das Fruchtfleisch mit ziviler und militärischer Kontrolle gleichgesetzt. So kann (in Nr. 3b) von der Lösung des Streites um die Orange auf die Lösung des Sinai-Konflikts geschlossen werden: Ägypten bekommt zivile und Israel militärische Kontrolle über Sinai (Baydin et al., 2011, S. 384). In diesem Beispiel wird der Orangen-Konflikt direkt als zuerst gefundener Referenzfall angegeben. Andere Quellen lassen den Mediator zunächst einen Konflikt um den Panamakanal wiedergeben, der als Lösung das Teilen in gleiche Teile ergibt. Weil Israel und Ägypten ein Halbieren der Sinai-Insel ablehnen, muss der Mediator in der Abfolge seiner Schritte zurückgehen und einen anderen Fall aus der Fallbasis herausuchen. (Kolodner and Simpson, 1989, S. 514-515)

3.7 Mediator: Retain

Im Falle der Lösungsakzeptanz wird die Fallbasis um den Fall c mit seiner Lösung S^* erweitert. Dies passiert allerdings nur, wenn der neue Fall sich mindestens um den vorher definierten und in einem anderen System evtl. anders festlegbaren Parameter θ von seinem analogen Fall unterscheidet.

$$if Sim(c, c^*) < \theta \quad (7)$$

Die Ähnlichkeit wird berechnet, um sicherzustellen, dass die Fallbasis signifikant bleibt oder Fälle sich sogar doppeln. Mit diesem Schritt endet die Mediation. Nach diesem Prinzip wird der Sinai-Konflikt mit seiner Lösung in die Falldatenbank des Mediators eingefügt, der Konflikt wurde gelöst.

4 Zusammenfassung und Fazit

Anhand der Einzelschritte des CBR-Zyklus und des Beispielfalls des Mediators wurde deutlich, dass die Entwicklung eines CBR-Systems an vielen Stellen variabel gehalten ist:

- der Umfang der Fallbasis und der Präzision des Indexvokabulars
- das Zuordnungsvermögen von passenden Fällen aus der Basis zu aktuellen Problemen (Retrieve)
- die Fähigkeit, alte Lösungen an neue Probleme anzupassen (Reuse)
- die Evaluation der entstandenen Lösung und die passende Korrektur (Retrieve)
- die Integration neuer Fälle in die Fallbasis (Retain) (Beierle and Kern-Isberner, 2014, S. 165-167)

Diese fünf Stellschrauben sind die Antwort auf die Forschungsfrage. In der Literatur ist vor allem die Adaptation umstritten, ob sie durch das System durchgeführt werden soll oder durch den Nutzer. Einigkeit besteht aber darüber, dass sie zu den schwierigsten Problemen eines CBR-Systems gehört. Insgesamt sind es aber die fünf Elemente des Systems, die gut durchdacht werden müssen, weil sie die Qualität des Systems beeinflussen (Beierle and Kern-Isberner, 2014, S. 203). Weiterführende Literatur zur Entwicklung eines eigenen Systems stellt die Metastudie von Bartsch-Spörl, Althoff und Meissonnier die 50 bereits existierende CBR-Systeme untersucht (Bartsch-Spörl and Althoff (2015)). Außerdem gibt es Programme wie myCBR (Bach et al. (2014)) und Colibri Studio (Recio-García et al. (2014)), die die Entwicklung unterstützen. Mit dem Lesen dieser Arbeit wurde der erste Schritt in die Richtung eines eigenen CBR-Systems gegangen. Wie im CBR-Zyklus folgen weitere Schritte, u. A. die wertvollen Maßnahmen Feedback und Lernen.

Literatur

- Kerstin Bach, Christian Sauer, Klaus-Dieter Althoff, and Thomas Roth-Berghofer. 2014. Knowledge Modeling with the Open Source Tool myCBR. (2014).
- B. Bartsch-Spörl and K.-D. Althoff. 2015. Commonsense Reasoning and Commonsense Knowledge in Artificial Intelligence. *Commun. ACM* 58, 9 (2015), 92–105.
- Atilim Guenes Baydin, Ramon Lopez de Mantaras, Simeon Simoff, and Carles Sierra. 2011. CBR with Commonsense Reasoning and Structure Mapping: An Application to Mediation. *Springer Berlin Heidelberg* 28 (2011), 378–392.
- Christoph Beierle and Gabriele Kern-Isberner. 2014. *Methoden wissensbasierter Systeme* (5nd ed.). Springer Vieweg, Wiesbaden.
- Ernest Davis and Gary Marcus. 2015. Commonsense Reasoning and Commonsense Knowledge in Artificial Intelligence. *Commun. ACM* 58, 9 (2015), 92–105.
- Dietrich Dörner. 1980. *Planen, Handeln und Entscheiden in sehr komplexen Realitätsbereichen* (1rst ed.). Hogrefe, Göttingen.
- Janet L. Kolodner. 1992. An introduction to case-based reasoning. *Artif. Intell. Rev.* 6, 1 (1992), 3–34.
- Janet L. Kolodner and Robert L. Simpson. 1989. The MEDIATOR: Analysis of an Early Case-Based Problem Solver. *Cognitive Science* 13, 4 (1989), 507–549.
- Juan A. Recio-García, Pedro A. González-Calero, and Belén Díaz-Agudo. 2014. Template-Based Design in COLIBRI Studio. *Information Systems* 40 (2014), 168 – 178.