Conflict Management for Constraint-based Recommendation[1]

Franz Wotawa, Martin Stettinger, Florian Reinfrank, Gerald Ninaus, Alexander Felfernig[2]
Institute for Software Technology, Graz University of Technology
Inffeldgasse 16b/II, 8010 Graz, Austria
{firstname.lastname}@ist.tugraz.at

Abstract

Constraint-based recommendation systems are well-established in several domains like cars, computers, and financial services. Such recommendation tasks are based on sets of product constraints and customer preferences. Customer preferences reduce the number of products which are relevant for the customer. In scenarios like that it may happen that the set of customer preferences is inconsistent with the set of constraints in the recommendation system. In order to repair an inconsistency, the customer is informed about possible ways to adapt his/her preferences. There are different possibilities to present this information to the customer: a) via preferred diagnoses, b) via preferred conflicts, and c) via similar products. On the basis of the results of an empirical study we show that diagnoses, conflicts, and similar products are evaluated differently by users in terms of understandability, user satisfaction, and conflict resolution effort.

1 Introduction

The number of e-commerce web sites and the quantity of offered products and services is increasing enormously [1]. This triggered the demand of intelligent techniques that improve the accessibility of complex item assortments for users. An approach to identify relevant products for each customer are recommendation systems [12]. We can differentiate between collaborative (e.g., www.amazon.com [12]), content-based (e.g., www.youtube.com [12]), critiquing-based (e.g., www.movielens.org [4]), and constraint-based systems (e.g., www.my-productadvisor.com [6]). The favored type of recommendation system depends on the domain in which the system will be used. For example, in highly structured domains where almost all information about a product is available in a structured form, constraint-based systems are often the most valuable recommendation approach.

Each recommendation approach has its own challenges. For example, collaborative systems have to deal with the cold-start problem [12]. For some content-based systems it is hard to identify related items [16]. Constraint-based systems can not offer products to users in each case. For example, a recommendation system contains notebooks from 2GB up to 8GB RAM but notebooks with 8GB RAM cost more than 1,000EUR. The user wants to buy a notebook with more than 8GB RAM at a price which is lower than 300EUR. The union of the product-related constraints with the customers preferences can not be fulfilled, i.e., the customer preferences are inconsistent with the given set of product-specific constraints. In such situations, a constraint-based recommendation system can provide help to users in terms of proposing change operations that restore the consistency between user preferences and the product-related constraints.

In this paper we present four different scenarios to support the user in finding a way out of the 'no solution could be found' dilemma. The first approach is to show which user preferences lead to an empty result set. For example, the combination of preferences of a notebook with 8GB RAM and a price which is lower than 200EUR is not satisfiable. By offering this information, the user can choose which of the preferences is less important and removes them. (1) We denote a set of preferences which is unsatisfiable (inconsistent with the given set of product-related constraints) as *conflict*. (2) Alternatively, the system is also able to show change operations which resolve all conflicts in the current customer preferences. Such change operations are denoted as *diagnoses*. (3) We can also show *diagnoses and explain them by giving the information about conflicts*. (4) If the user is not interested in conflicts and diagnoses, we are also able to show similar products by using a *utility function* which is ranking the products according to the user's preferences.

The major goal of this paper is to analyze, in which way inconsistencies should be presented to users. Therefore, we conducted a study at the Graz University of Technology and the University of Klagenfurt. With this empirical study we provide recommendations for presenting inconsistencies in constraint-based recommendation scenarios to users.

The remainder of this paper is organized as follows. Section 2 gives an introduction into constraint-based recommendation systems, shows a working example, and provides an overview of inconsistency management techniques and utility calculation for products. Section 3 shows our online application for the empirical study, lists our hypotheses, and shows the evaluation of the hypotheses. Section 4 discusses relevant aspects and Section 5 finalizes this paper with a summary and issues for future work.

[1]We thank the anonymous reviewers for their helpful comments.
[2]Authors are ordered in reverse alphabetical order.

2 Constraint-based recommendation systems

In our approach we exploit constraint satisfaction problems (CSPs) for representing products and customer preferences [20]. Such CSPs are a major modeling technique for knowledge bases [3; 7]. CSPs are represented by a triple (V, D, C) where V is a set of variables (see the following example). $V = \{v_{name}, v_{CPU}, v_{RAM}, v_{HDD}, v_{LCD}, v_{price}\}$

D is a set of domains $dom(v_i)$ where each domain describes possible assignments for a variable, for example:
$D = \{\ dom(v_{name}) = \{cheap, media, easy, turbo\},$
$dom(v_{CPU}) = \{dualcore, quadcore\},$
$dom(v_{RAM}) = \{4GB, 6GB, 8GB\},$
$dom(v_{HDD}) = \{400GB, 500GB, 750GB\},$
$dom(v_{LCD}) = \{14, 15, 17\},$
$dom(v_{price}) = \{199, 399, 599\}\}$

The set C describes constraints which are reducing the product space. Constraints can be *product constraints* $c \in C_{KB}$ and all products are combined in a disjunctive order, such that $c_0 \vee c_1 \vee ... \vee c_n \in C_{KB}$. A conjunctive set of *customer preferences* $c \in C_R$ describes the customers preferences and $C_{KB} \wedge C_R = C$. Next, we insert four products into C_{KB} and two customer preferences into C_R.

$c_0 : v_{name} = cheap \wedge v_{CPU} = dualcore \wedge v_{RAM} = 4GB$
$\wedge v_{HDD} = 400 \wedge v_{LCD} = 15 \wedge v_{price} = 199$
$c_1 : v_{name} = media \wedge v_{CPU} = dualcore \wedge v_{RAM} = 8GB$
$\wedge v_{HDD} = 750 \wedge v_{LCD} = 17 \wedge v_{price} = 599$
$c_2 : v_{name} = easy \wedge v_{CPU} = quadcore \wedge v_{RAM} = 4GB$
$\wedge v_{HDD} = 500 \wedge v_{LCD} = 14 \wedge v_{price} = 399$
$c_3 : v_{name} = turbo \wedge v_{CPU} = quadcore \wedge v_{RAM} = 8GB$
$\wedge v_{HDD} = 750 \wedge v_{LCD} = 15 \wedge v_{price} = 599$
$c_4 : v_{CPU} = dualcore \in C_R$
$c_5 : v_{RAM} \geq 6GB \in C_R$

We now try to get all valid instances of the constraint-based recommendation task. A result (solution or instance) of such a recommendation task is characterized by Definition 1.

Definition 1: A complete consistent *instance* is a model where each variable in the knowledge base has an assignment, i.e. $\forall_{v \in V} v \neq \emptyset$ and all assignments are consistent with the constraints in C.

In our case, the product c_1 fits all customer constraints (preferences). Now, let's assume that the customer has one more preference and adds the following constraint:

$c_6 : v_{LCD} = 14 \in C_R$

The new recommendation task leads to an inconsistency, s.t. Definition 1 cann't be fulfilled. We only consider the constraints in C_R as conflicting constraints and assume that the products in C_{KB} have a valid representation.

Definition 2: A *conflict set* is a set of constraints $CS \subseteq C_R$ s.t. $CS \subseteq C_{KB}$ is inconsistent.

In the example, C_R is inconsistent with C_{KB}. Because potentially non-minimal conflict sets are not helpful for users, we try to reduce the number of constraints in conflict sets CS as much as possible (see Definition 3) and introduce the term minimal conflict set.

Definition 3: A *minimal conflict set* is given, iff the set CS is a minimal conflict set (see Definition 2) and there does not exist a conflict set $CS' \subset CS$ with the property of being a conflict set.

There are algorithms for calculating minimal conflict sets in inconsistent constraint sets [13]. In our scenario, a conflict set detection algorithm calculates the set $CS = \{c_4, c_6\}$ as a minimal conflict set. Since conflict detection algorithms typically return one minimal conflict set at a time (see, e.g., Junker [13]), we use Reiter's HSDAG to calculate all minimal conflicts [17]. In our example we get two different minimal conflict sets: $CS_1 = \{c_4, c_6\}, CS_2 = \{c_5, c_6\}$.

Not all constraint sets have the same importance for each user. For example, if the CPU is more important for a user than the RAM, the user will probably prefer conflict sets which do not contain the CPU. We can calculate *preferred minimal conflicts* by ordering the constraints s.t. the preferred constraints are at the end of the list. For example, the constraint ordering $\{c_6, c_5, c_4\}$ leads to the conflict set $\{c_6, c_5\}$ [9].

Resolving conflicts can be done in two different ways: First, we can remove constraints from a conflict set and receive further conflicts (e.g., removing the constraint c_4 leads to the conflict set $CS = \{c_5, c_6\}$). Second, we can determine a set of constraints which resolves all conflicts in the given set of user preferences. We denote such sets diagnoses [8; 17]. By removing a set of constraints Δ from the set of user preferences, we receive at least one valid instance (solution). A formal definition of diagnosis is the following one (see Definition 4):

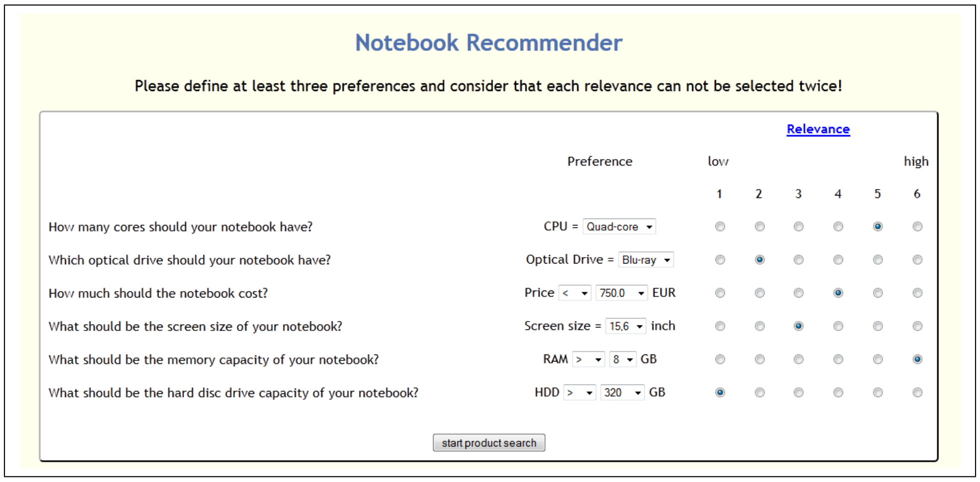
Definition 4: A set of constraints $\Delta \subseteq C_R$ is denoted as diagnosis if $C_R \setminus \Delta \cup C_{KB}$ is consistent.

In our example, the removal of the set C_R restores consistency since C_{KB} is consistent. As the removal of all constraints probably doesn't satisfy the customer, we try to detect minimal sets of diagnoses (see Definition 5) which will be used in Scenarios 1 and 3 as explanations for the inconsistency (Scenario 3 uses the minimal conflicts as an explanation of the diagnoses).

Definition 5: A set of constraints $\Delta \subseteq C_R$ is denoted as *minimal diagnoses* iff it is a diagnosis (see Definition 4) and there does not exist a diagnosis Δ' with $\Delta' \subset \Delta$.

The example notebook recommendation system contains two different minimal diagnoses: $\Delta_1 = \{c_4, c_5\}, \Delta_2 = \{c_6\}$. We can calculate them by using a diagnosis detection algorithm [19] within the HSDAG for calculating all possible diagnoses [17] and order the diagnoses based on the ordering of the constraints in C_R [9].

Next, we calculate all minimal conflicts (Scenarios 2 and 3) and minimal diagnoses (Scenarios 1 and 3) for each customer. Currently it is not considered which of the conflict and diagnoses sets contains preference constraints that are relevant for the customer. For example, if the CPU is more important for the customer than the LCD size and the RAM (i.e., $relevance(v_{CPU}) = 3$, $relevance(v_{LCD}) = 2, relevance$ $(v_{RAM}) = 1$) we order the conflicts and diagnoses based on the relevance of the customer preferences. A conflict / diagnosis containing low relevances is called a *preferred minimal conflict / diagnosis* [8; 10]. In our example, the user has the possibility to add a relevance for each customer constraint $1 \leq relevance(v_i) \leq n$ where n is the number of all variables. We used this information in our empirical study to get preferred minimal conflicts and preferred minimal diag-


Notebook Recommender

Please define at least three preferences and consider that each relevance can not be selected twice!

	Preference	1 (low)	2	3	4	5	6 (high)
How many cores should your notebook have?	CPU = Quad-core	○	○	○	○	●	○
Which optical drive should your notebook have?	Optical Drive = Blu-ray	○	●	○	○	○	○
How much should the notebook cost?	Price < 750.0 EUR	○	○	○	●	○	○
What should be the screen size of your notebook?	Screen size = 15,6 inch	○	○	●	○	○	○
What should be the memory capacity of your notebook?	RAM > 8 GB	○	○	○	○	○	●
What should be the hard disc drive capacity of your notebook?	HDD > 320 GB	●	○	○	○	○	○

Figure 1: Notebook recommendation: definition and weighting of user preferences. Each relevance can only be selected once.

noses. For a detailed discussion of algorithms supporting the determination of preferred conflicts and diagnoses we refer the reader to Felfernig and Schubert [8].

We are also able to evaluate similarities between products and the customer preferences which will be used in the fourth Scenario of our empirical study. If the customer preferences can not be fulfilled, we can calculate the similarity by using the fitness function given in Equation 1.

$$fit(p, C_R) = \sum_{c \in C_R} u(p, c) \times \omega(max_{relevance}, c) \qquad (1)$$

In Equation 1, p defines a product. C_R is the set of customer preferences. For each customer preference we calculate the utility value $u(p, c)$ and the weighting $\omega(max_{relevance}, c)$. For the utility value, we are using McSherrys' similarity metrics for each variable [15]. For example, a lower price value is better (less is better) $\frac{customer\_value}{product\_value}$, a higher RAM value is better (more is better) $\frac{product\_value}{customer\_value}$ and for the optical drive a nearer value is better (nearer is better) $= [0, 1]$. The weighting function $\omega(max_{relevance}, c)$ evaluates a weighting for the constraint c by calculating the relative importance $\frac{relevance(c)}{max_{relevance}}$. In the example in Figure 1 the weighing function for the product variable CPU is $5/6$. Table 1 gives an overview about the fitness values of all example products (see Section 2). Note that the application upgraded all fitness values to a percentile value. The best value was the number of fulfilled preferences divided by the number of all the user's preferences. In our example the product c_1 matches two of three preferences ($\frac{2}{3} = 66\%$). The second best value was devaluated by the relative difference between the fitness values ($0.66\frac{0.460}{0.535} = 57\%$).

3 Empirical Study

How users of recommendation systems deal with conflicts, diagnoses, and fitness values will be evaluated in this section. Therefore, we describe our online notebook recommendation system, define hypotheses, and evaluate and discuss them based on an empirical study.

product	fitness	percentile
c_0	0.460	57%
c_1	0.535	66%
c_2	0.222	27%
c_3	0.303	37%

Table 1: Fitness values for the example knowledge base. For example, for the product c_0 and the customer preferences $C_R = \{c_4, c_5, c_6\}$ the fitness value is calculated by $(1 \times \frac{3}{3}) + (\frac{4}{6} \times \frac{1}{3}) + (\frac{14}{15} \times \frac{2}{3})$.

3.1 Notebook Recommendation System

In the preferences screen (see Figure 1) the user is asked for at least three preferences which are described in terms of product variables. Each of the specified preferences must be weighted on a six-point scale.

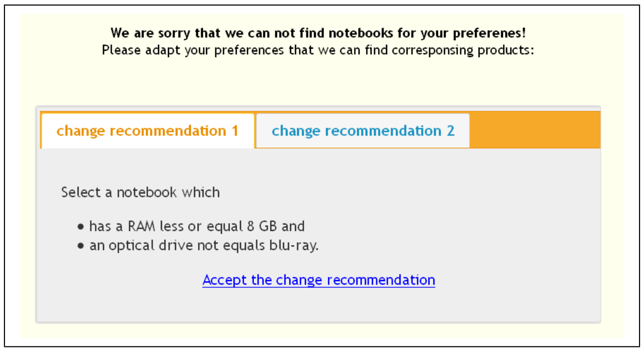
The next step was to remove all products $c \in C_{KB}$ which are consistent with the user preferences C_R to assure, that the participants were confronted with a situation where her preferences were inconsistent with the underlying product assortment, i.e., C_R is inconsistent with C_{KB}.

In the following, participants received a visualization of the conflict. Each participant was assigned to one of four scenarios (see Table 2). In the first scenario the participants got *minimal diagnoses* as change recommendations (see Figure 2). Scenario 2 presents *minimal conflicts* to the participants (see Figure 3). Scenario 3 contains both, *minimal diagnoses and minimal conflicts*, as explanations (see Figure 4). Scenario 4 shows the fitness values for all products (see Figure 5). For the differentiation between experts and novices we used two questions in the questionnaire at the end of the study. The first question asked for a self-assessment and the second question asked for expert knowledge. In our study 111 participants are experts and 90 participants are novices.

Next, we try to find the best approach for presenting inconsistencies in constraint-based recommendation systems. For the evaluation we have measured three general characteristics: a) the *time* which is used to repair a conflict, b) the

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Step 1	Insert preferences			
Step 2	apply diagnoses	dissolve conflicts	apply diagnoses	
Step 3	Select a product			
Step 4	Answer a questionaire			

Table 2: Overview about the user activities and scenarios

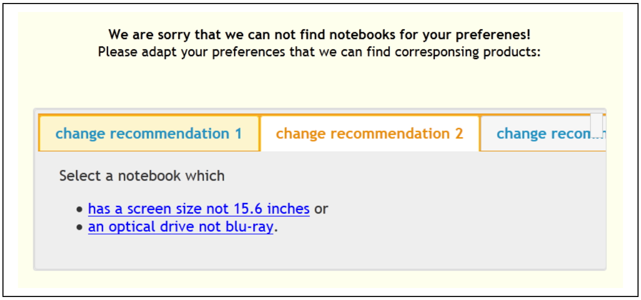

We are sorry that we can not find notebooks for your preferenes!
Please adapt your preferences that we can find corresponsing products:

Select a notebook which

- has a RAM less or equal 8 GB and
- an optical drive not equals blu-ray.

Accept the change recommendation

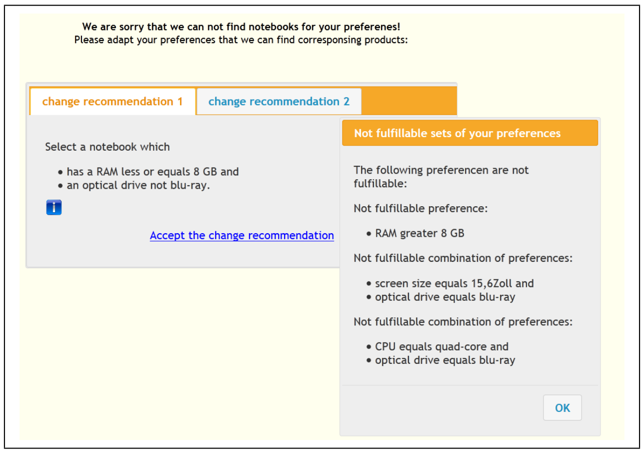
Figure 2: Presentation of 1 to n diagnoses.


We are sorry that we can not find notebooks for your preferenes!
Please adapt your preferences that we can find corresponsing products:

Select a notebook which

- has a screen size not 15.6 inches or
- an optical drive not blu-ray.

Figure 3: Presentation of 1 to n conflicts.


We are sorry that we can not find notebooks for your preferenes!
Please adapt your preferences that we can find corresponsing products:

Select a notebook which

- has a RAM less or equals 8 GB and
- an optical drive not blu-ray.

Accept the change recommendation

Not fulfillable sets of your preferences

The following preferencen are not fulfillable:

Not fulfillable preference:

- RAM greater 8 GB

Not fulfillable combination of preferences:

- screen size equals 15,6Zoll and
- optical drive equals blu-ray

Not fulfillable combination of preferences:

- CPU equals quad-core and
- optical drive equals blu-ray

Figure 4: Presentation of 1 to n diagnoses and conflicts.

understandability of conflicts and diagnoses, and c) the *satisfaction* with the 'no solution could be found' dilemma.

3.2 Hypotheses

After having selected a diagnosis (in Scenarios 1 and 3), the participant (user) receives a list of notebooks. In Scenario 2 the user has to remove as many of her preferences unless a product can be recommended because we removed all products which fits to the preferences. We call the number of preferences, which have to be removed until the user receives products, interaction cycles. For example, an interaction cycle of two means that the user removed two of her preferences until products could be presented. Therefore we expect that the time, which is necessary for resolving the conflict, will be lower when diagnoses are presented to the participant:

Hypothesis 1: Study participants will solve inconsistencies faster when they receive diagnoses.

The study participants received all diagnoses in a preferred order (see Section 2). We expect that the first diagnosis will be selected most frequently.

Hypothesis 2: The first diagnosis will be selected by the majority of the users for adapting their preferences.

A conflict occurs if a set of preferences can not be fulfilled (see Definitions 2 and 3). Scenario 3 uses the minimal conflict sets (see Definition 3) as a description for the minimal diagnoses (see Definition 5). We expect a positive impact on the understandability by the diagnoses:

Hypothesis 3: Participants will understand their conflicts more easily, if they receive explanations.

When the participants don't receive products after having inserted the preferences, the satisfaction with the recommendation system will decrease, and we expect that the satisfaction with the product assortment of our recommendation system will be higher if products are offered (Scenario 4), even if they don't fulfill all of the participants' preferences.

Hypothesis 4: The participants will have a higher satisfaction with the product assortment when they receive fitness values (Scenario 4, see Figure 5).

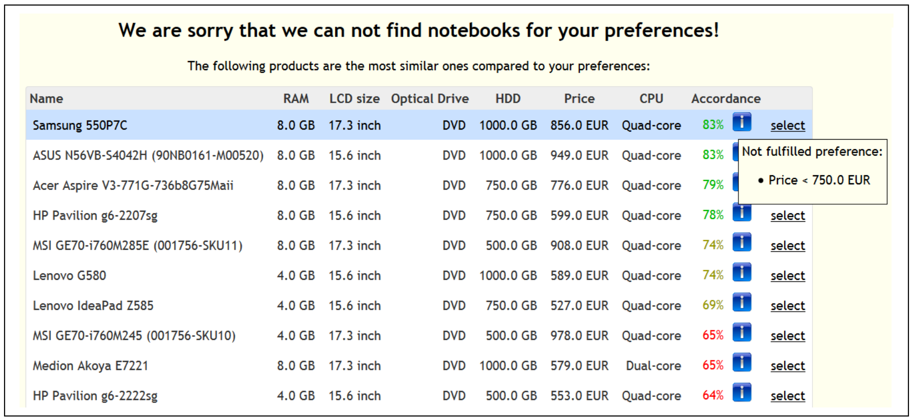
Due to the stability of preferences, the participants are less willing to adapt their preferences. When the recommendation system asks for more than one adaption of preferences, the participants will have a lower satisfaction with the system. This leads to the following Hypothesis:

Hypothesis 5: More interaction cycles lead to a lower satisfaction with the anomaly support.

3.3 Evaluation

For evaluating our hypotheses, we conducted a study at the TU Graz and the University of Klagenfurt. 240 users participated in our study. The students' average age is 25 years (std. dev.: 5.52 years). The participants are studying technical sciences (117), cultural sciences (63), economics (29), and other sciences (n = 31). We've tested our results with a two-tailed Mann-Whitney U-test and removed all participants with contradictory answers to the SUS (system usability scale) questionnaire [2]. Finally, we divided 201 participations into the scenarios with diagnoses (n = 56), conflicts (n = 50), diagnoses and conflicts (n = 38), and the fitness function (n = 57).

Hypothesis 1 focuses on the time which is required to resolve inconsistencies. Therefore, we measured the time between the first conflict notification and the product presentation (see Table 3).



We are sorry that we can not find notebooks for your preferences!

The following products are the most similar ones compared to your preferences:

Name	RAM	LCD size	Optical Drive	HDD	Price	CPU	Accordance	
Samsung 550P7C	8.0 GB	17.3 inch	DVD	1000.0 GB	856.0 EUR	Quad-core	83%	select
ASUS N56VB-S4042H (90NB0161-M00520)	8.0 GB	15.6 inch	DVD	1000.0 GB	949.0 EUR	Quad-core	83%	
Acer Aspire V3-771G-736b8G75Maii	8.0 GB	17.3 inch	DVD	750.0 GB	776.0 EUR	Quad-core	79%	
HP Pavilion g6-2207sg	8.0 GB	15.6 inch	DVD	750.0 GB	599.0 EUR	Quad-core	78%	select
MSI GE70-i760M285E (001756-SKU11)	8.0 GB	17.3 inch	DVD	500.0 GB	908.0 EUR	Quad-core	74%	select
Lenovo G580	4.0 GB	15.6 inch	DVD	1000.0 GB	589.0 EUR	Quad-core	74%	select
Lenovo IdeaPad Z585	4.0 GB	15.6 inch	DVD	750.0 GB	527.0 EUR	Quad-core	69%	select
MSI GE70-i760M245 (001756-SKU10)	4.0 GB	17.3 inch	DVD	500.0 GB	978.0 EUR	Quad-core	65%	select
Medion Akoya E7221	8.0 GB	17.3 inch	DVD	1000.0 GB	579.0 EUR	Dual-core	65%	select
HP Pavilion g6-2222sg	4.0 GB	15.6 inch	DVD	500.0 GB	553.0 EUR	Quad-core	64%	select

Not fulfilled preference:

- Price < 750.0 EUR

Figure 5: Presentation of fitness values.

Scenario	1 D	2 C	3 D&C	4 Fit
conflict solving time	16.64	21.16	20.05	0.00
product selection time	26.09	27.52	18.72	43.82
total	42.73	48.68	38.77	43.82

Table 3: Average time (in sec.) to resolve inconsistencies and to select a product (in sec.; D = diagnoses, C = conflicts, Fit = fitness)

The result shows that the time for removing conflicts with diagnosis is lower (16.64 sec.) than with conflicts (21.16 sec.) or selecting the diagnoses with a corresponding explanation (20.05 sec.). This is because there is only one interaction cycle for resolving inconsistencies with a diagnosis whereas 1.66 interaction cycles are required to resolve inconsistencies with conflicts. Reading the explanation of a diagnosis also increased the time to resolve an inconsistency (20.05 sec.) compared to the diagnoses without explanations ($p < 0.1$). The time for resolving the conflicts is 0 in Scenario 4 since they aren't resolved. **These results confirm Hypothesis 1**.

We also researched the influences of the number of conflicts and diagnoses (see Table 4).

# of presented diagnoses / conflicts	n	satisfaction	repair time
1 diagnosis:	11	4.55	11.18 sec.
2 diagnoses:	11	4.14	10.71 sec.
> 2 diagnoses:	38	4.37	19.32 sec.
1 conflict:	56	4.09	22.29 sec.
2 conflicts:	23	4.04	45.48 sec.
>2 conflicts:	4	1.75	62.00 sec.

Table 4: Average time to repair conflicts regarded to the number of presented conflicts

The time to select a product was nearly the same in the scenarios with diagnoses (Scenario 1) and conflicts (Scenario 2). The third scenario performs best in terms of the time which is required to select a product (18.72 sec.). This can be explained by the fact that dealing with diagnoses and conflicts helps to receive a deep understanding of the problem. Participants in the fourth Scenario required 43.82 sec. for selecting a product. The higher effort for selecting a product can be explained by the missing explanations of the conflict, and the participants may get confused that not all preferences are fulfilled by the offered products. All differences in the product selection time are statistically significant ($p < 0.001$).

Hypothesis 2 is looking at the ordering of preferred diagnoses and conflicts. We measured the position of the selected conflict / diagnoses (see Figure 6). Note, there are only those participants considered from Scenarios 1 and 3 whose number of offered diagnoses is greater than one.

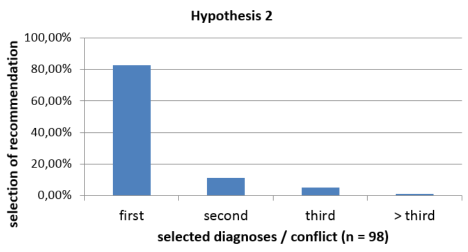


Figure 6: Ranking of selected diagnosis / conflict

We can confirm Hypothesis 2 since 81 of the participants (82.65%) selected the first diagnosis. The second diagnosis was selected by 11 (11.22%), the third one by 5 (5.10%) participants and the fourth recommendation by one participant (1.02%). Reasons for applying the first diagnosis are that the first diagnosis contains only unimportant preferences, the primacy-effect [5], and preference reversals [22].

For measuring **Hypothesis 3** we asked the participants from the Scenarios 1-3 if the diagnoses/conflicts were understandable. Answers were given on a 5 point Likert-scale (5 represents the highest understandability).

Results show that the highest understandability is given when diagnoses are presented (Scenario 1) followed by diagnoses explained with conflicts (Scenario 3) and conflicts (Scenario 2, see Table 5). The difference between the understandability of conflicts (4, 4, Scenario 2) and the other

Scenarios (Scenario 1 with 4.55 and Scenario 3 with 4.45) is statistically significant ($p < 0.05$). The degree of understandability is higher for experts than for novices ($p > 0.1$). We can **partially confirm Hypothesis 3** since experts have a higher understanding of the conflict when conflicts and diagnoses are presented while novices can not deal with much information. Due to the cognitive processes (trial-and-error of novices versus analytical processing of experts [11]) it is easier to deal with diagnoses when the cognitive process is more analytical. When participants use a trial-and-error process and they don't expect the visualization of conflicts, it is harder to adapt the preferences.

Scenario	1 D	2 C	3 D&C
Total:	4.55	4.40	4.45
Experts:	4.62	4.38	4.67
Novices:	4.46	4.42	4.18

Table 5: Understandability of conflicts

Hypothesis 4 evaluates the satisfaction with the recommended products. The average values are from 2.62 up to 3.3 (see Table 6) which is worse and can be explained by the removal of all valid products at the beginning of the process.

Scenario	1 D	2 C	3 D&C	4 Fit
Total	2.62	3.30	2.80	3.30
Experts	2.44	3.12	2.33	3.19
Novices	2.88	3.50	3.35	3.48

Table 6: Satisfaction with the product assortment

The results show that conflicts (Scenario 2) and the fitness function (Scenario 4) lead to the highest satisfaction with the product assortment. A differentiation between experts and novices does not influence the significance. Because conflicts and the fitness values lead to the same satisfaction **we can not confirm Hypothesis 4**. An interesting result is also, that novices have an overall higher satisfaction with the product assortment compared to experts. This can be explained by the fact, that they are more happy that they get any products recommended. On the other hand, experts know, that there are products which fits to their preferences.

Hypothesis 5 will be evaluated by Table 7. There is a significant difference when participants had more than two interaction cycles. A statistically significant difference between experts and novices isn't constituted. A differentiation between the interaction cycles of diagnoses and conflicts also doesn't lead to a significant difference between all interaction cycles or between conflict and diagnoses visualization. **We can confirm Hypothesis 5**.

A comparison between the number of conflicts / diagnoses and satisfaction, understandability, or time to resolve the inconsistency is not statistically significant.

# interaction cycles	n	satisfaction
1	34	4.44
2	10	4.30
3	3	2.67
≥ 4	3	3.00

Table 7: Satisfaction with the presented conflicts regarding to interaction cycles

4 Discussion

This paper gives an overview about conflict management in constraint-based recommendation systems. While we can not present products which fit to the user's preferences the user has to adapt her preferences. Such preference reversals always result in a low satisfaction of users. The degree of dissatisfaction depends on how often the preferences have been fulfilled in the past [22].

If users have positive experience with their preferences, it can happen that the participants have well-established anchoring affects [21]. In such scenarios the participants may have stable preferences and preference reversals are necessary to get notebooks. It can be more problematic if there are many conflicts / diagnoses shown, because it could be the case that a representation of all conflicts / diagnoses leads to a manifestation of the current preferences and the user is less willing to accept any conflicts / diagnoses. Such an effect is called status-quo bias [14; 18].

Another important aspect is the cognitive processing task. While novices tend to use trial-and-error processes, experts tend to use heuristic and analytic cognitive processes [11]. That means that novices tend to adapt their preferences unless they receive products. Our results confirm this process since the satisfaction of novices is high if they can adjust their preferences arbitrarily or receive similar products (see Hypothesis 4). On the other hand, experts try to understand the modifications and analyze them. Therefore, they prefer the visualization of diagnoses (see Hypothesis 3).

5 Conclusion

This paper shows how different visualization strategies for conflicts can be used within constraint-based recommendation systems. We've shown the state-of-the-art in detecting all minimal preferred diagnoses and conflicts, calculated fitness values for products, and introduced hypotheses for conflict management and evaluated them with an empirical study. The result of this evaluation is that the optimal strategy for the visualization of inconsistencies depends on the optimization strategy. The visualization of diagnoses leads to a low interaction effort, whereas the visualization of conflicts and fitness functions leads to a higher satisfaction.

A major focus of our future work will be the inclusion of different decision-psychological effects such as, for example, framing, priming, and decoy effects into our studies. In this context we want to answer the question whether these phenomena exist in the context of conflict detection and resolution scenarios, too.

References

[1] Ivan Arribas, Francisco Perez, and Emili Tortosa-Ausina. Measuring international economic integration: Theory and evidence of globalization. *World Development*, 37(1):127 – 145, 2009.

[2] John Brooke. Sus: a quick and dirty usability scale. In Patrick Jordan, B. Thomas, Bernard Weerdmeester, and Ian Lyall McClelland, editors, *Usability Evaluation in Industry*. Taylor and Francis, 1986.

[3] Robin Burke. Knowledge-based recommender systems. In *Encyclopedia of library and information systems*, page 2000. Marcel Dekker, 2000.

[4] Li Chen and Pearl Pu. Evaluating critiquing-based recommender agents. In *Proceedings of the 21st national conference on Artificial intelligence - Volume 1*, AAAI'06, pages 157–162. AAAI Press, 2006.

[5] Alexander Felfernig et al. Persuasive recommendation: Serial position effects in knowledge-based recommender systems. In Yvonne Kort, Wijnand IJsselsteijn, Cees Midden, Berry Eggen, and B.J. Fogg, editors, *Persuasive Technology*, volume 4744 of *Lecture Notes in Computer Science*, pages 283–294. Springer Berlin Heidelberg, 2007.

[6] Alexander Felfernig and Robin Burke. Constraint-based recommender systems: technologies and research issues. In *Proceedings of the 10th international conference on Electronic commerce*, ICEC '08, pages 3:1–3:10, New York, NY, USA, 2008. ACM.

[7] Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Stumptner. Consistency-based diagnosis of configuration knowledge bases. *Artificial Intelligence*, 152(2):213 – 234, 2004.

[8] Alexander Felfernig and Monika Schubert. Personalized diagnoses for inconsistent user requirements. *AI EDAM*, 25(2):175–183, 2011.

[9] Alexander Felfernig, Monika Schubert, and Stefan Reiterer. Personalized diagnosis for over-constrained problems. *IJCAI*, pages 1990 – 1996, 2013.

[10] Alexander Felfernig, Monika Schubert, and Christoph Zehentner. An efficient diagnosis algorithm for inconsistent constraint sets. *AI EDAM*, 26(1):53–62, 2012.

[11] Jon-Chao Hong and Ming-Chou Liu. A study on thinking strategy between experts and novices of computer games. *Computers in Human Behavior*, 19:245 – 258, 2003.

[12] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems: An Introduction*, volume 1. University Press, Cambridge, 2010.

[13] Ulrich Junker. Quickxplain: preferred explanations and relaxations for over-constrained problems. In *Proceedings of the 19th national conference on Artifical intelligence*, AAAI'04, pages 167–172. AAAI Press, 2004.

[14] Daniel Kahneman, Jack Knetsch, and Richard H. Thaler. Anomalies: The endowment effect, loss aversion, and status quo bias. *The Journal of Economic Perspectives*, 5:193 – 206, 1991.

[15] David McSherry. Similarity and compromise. In *Proceedings of the Fifth International Conference on Case-Based Reasoning*, pages 291–305. Springer, 2003.

[16] Michael Pazzani and Daniel Billsus. Content-based recommendation systems. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 325–341. Springer Berlin / Heidelberg, 2007.

[17] Raymond Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, 1987.

[18] William Samuelson and Richard Zeckhauser. Status quo bias in decision making. *Journal of Risk and Uncertainty*, 1:7–59, 1988.

[19] Monika Schubert and Alexander Felfernig. A Diagnosis Algorithm for Inconsistent Constraint Sets. In *Proceedings of the 21st International Workshop on the Principles of Diagnosis*, 2010.

[20] Edward Tsang. *Foundations of Constraint Satisfaction*. Academic Press, 1993.

[21] Amos Tversky and Daniel Kahneman. Judgement under uncertainty: Heuristics and biases. *Science*, 185(4157):1124 – 1131, 1974.

[22] Amos Tversky, Paul Slovic, and Daniel Kahneman. The causes of preference reversal. *American Economic Review*, 80(1):204–17, March 1990.